University of Alberta

ADMISSION CONTROL OF DELAY BOUNDED TRAFFIC IN CELLULAR NETWORKS

by

Yaser Khamayseh

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Doctor of Philosophy**.

Department of Computing Science

Edmonton, Alberta

Fall 2007

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-32995-5
Our file Notre référence
ISBN: 978-0-494-32995-5

NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Abstract

In this thesis we consider problems arising in delivering delay bounded traffic to mobile users of cellular networks. Such traffic is important to support (for example, for delivering multimedia streaming services), yet challenging to manage, since it typically requires relatively high data rates that consume significant wireless system resources. Moreover, users expect uninterrupted operation while roaming within the coverage area.

The problems formalized in the thesis are investigated in the context of managing two types of cellular networks:

- networks that support multi user access through time division multiplexing where a fixed number of channels is allocated to serve the streaming requests, and
- networks that support multi user access through code division multiple access (CDMA). Such networks are characterized by a soft capacity aspect.

For networks of the first type, we devise admission control (CAC) mechanisms that keep track of network state at any instant by utilizing scheduling mechanisms that take into account delay constraints on individual traffic connection requests. Two types of scheduling mechanisms are considered in the thesis for the above purpose: non-preemptive scheduling that assumes that a connection request is served to completion without service interruption, and preemptive scheduling that aims at achieving higher throughput by allowing service preemption. In both cases, the thesis develops frameworks for the devised CAC and the underlying scheduling mechanism, present quantitative analysis of the designed schedulers, and evaluate the performance of the devised frameworks by simulation. A novel contribution

of the thesis is the design and analysis of CAC architectures for the first type of networks to serve delay bounded traffic.

For networks of the second type, we devise a CAC mechanism that keeps track of network state at any instant by keeping track of both intra-cell and inter-cell mobility of served users in order to estimate the cell overload probability after a prediction interval in the future. Such CAC architecture has been devised in the literature for rate sensitive (but not particularly delay bounded) traffic. A novel aspect of the thesis is on extending the architecture to our present context of serving delay bounded traffic in soft capacity networks.

To my family and friends

Acknowledgements

I would like to thank my advisor, Dr. Ehab Elmallah for his constant help and support during my study. I thank my committee memebers, Dr. Mike MacGregor, Dr. Ioanis Nikolaidis, and Dr. Janelle Harms for their constructive feedback and suggestions. I would also like to thank Dr. Azzedine Boukerche from university of Ottawa, and Dr. Mrinal Mandal for participating in my dissertation committee.

Table of Contents

1 Introduction **1**

1.1 Introduction 2

1.2 Background on Some Cellular Networks Architectures 3

1.3 Power and Interference Aspects in Wireless Networks 5

1.4 Resource Management in Cellular Networks 10

1.5 Issues with User Mobility 11

1.6 Traffic Modeling 15

1.7 Issues with Multimedia Encoding Techniques 16

1.8 Multimedia Streaming 18

1.9 Summary 20

2 Literature Review **21**

2.1 Admission Control for Voice Traffic 22

2.2 Admission Control and Scheduling for Mixed Voice and Data Traffic 24

2.3 Scheduling Algorithms 26

2.3.1 Task Scheduling Problems 26

2.3.2 Packet Scheduling Problems 27

2.4 QoS Provisioning using an Adaptive Multimedia Framework 30

2.4.1 System Model 31

2.4.2 Call Admission Control 31

2.4.3 The Bandwidth Allocation Algorithm (BAA) 33

2.4.4 Summary 33

2.5 Downlink Scheduling in CDMA Data Network 34

2.5.1 Performance Measures and Results 35

2.5.2 Summary 37
2.6 Dynamic Bandwidth Allocation with Fair Scheduling for WCDMA Networks 37
2.6.1 CDGPS Fair Scheduling Scheme 37
2.7 Results 39
2.8 Summary 39

3 Admission Control of Delay Bounded Traffic Using Non-preemptive Scheduling 40
3.1 Introduction 41
3.2 System Model 42
3.3 Problem Formulation 44
3.4 Background Results and Remarks 47
3.5 Call Admission Control Framework 48
3.6 T_{plan} Adapter (DBAS Front End) 49
3.7 The Scheduler (DBAS Back End) 53
3.7.1 Design of k-Channel Packing Schedulers 53
3.7.2 Scheduling Algorithm 55
3.7.3 Two-Channel Packing Scheduler 55
3.7.4 One-channel Packing Scheduler 60
3.8 Performance Results 61
3.8.1 Evaluating the Exponential Moving Average Effect 62
3.8.2 Evaluating the Scheduler Performance 62
3.8.3 Evaluating the Overall Framework Performance 68
3.9 Concluding Remarks 69

4 Admission Control of Delay Bounded Traffic Using Preemptive Scheduling 70
4.1 Introduction 71
4.2 System Model and Problem Formulation 72
4.3 Framework Architecture 77
4.3.1 The T_{plan} Adapter 77

4.3.2 The Scheduler 79
4.4 Performance Results 85
4.4.1 Time-Slotted Predictive CAC 85
4.4.2 Simulation Environment 86
4.4.3 Numerical Results 88
4.5 Concluding Remarks 91

5 Admission Control of Delay Bounded Traffic for CDMA Networks 92
5.1 Introduction 93
5.2 System Model 95
5.3 Call Admission Control Architecture 98
5.3.1 Physical Layer Constraints 100
5.3.2 Admission Procedure 101
5.4 Overload Probability Computation 102
5.4.1 Two-way Symmetric Diagram Overload Probability Calculations 103
5.4.2 Two-way Symmetric Diagram Overload Probability Algorithm 105
5.5 Scheduling Algorithm 111
5.6 Performance Results 112
5.6.1 Description of CAC Schemes Used for Comparison 113
5.6.2 Simulation Parameters 117
5.6.3 Numerical Results 117
5.7 Concluding Remarks 122

6 Conclusion and Future Work 123
6.1 Summary 123
6.2 Future Work 124

List of Tables

1.1 Mathematical Symbols Summary Table 10

2.1 Summary of Mathematical Symbols Used in [46] 32
2.2 Summary of Mathematical Symbols Used in [39] 34
2.3 Summary of Mathematical Symbols Used in [80] 38

3.1 Simulation Parameters 65

4.1 Simulation Parameters 87

5.1 Important Admission and Overload Calculation Procedures Parameters 106
5.2 Parameters and Associated Values 118

List of Figures

1.1 UMTS Network Architecture 4
1.2 Two Tier Cell Structure Model [77] 9
1.3 Example of Multimedia Streaming Flow 19

3.1 A Scenario with 3 Connection Requests (dashed lines indicate allowable delays). 48
3.2 Pseudo-code of the T_{plan} Adapter of the Delay Bounded Adaptive Scheduling (DBAS) Framework. 52
3.3 A Scenario with Five Connection Requests. 54
3.4 Pseudo-code of the Scheduler of the DBAS Framework. 55
3.5 The set DR of all Undelayed, and Delayed Connection Request Instances of the Set R in Figure 3.3 (a total of $2+5+2+3+7=19$ instances). 58
3.6 Pseudo-code for Function 2-CPS. 59
3.7 Effect of Choosing the Parameter α in the EMA on the Effective Throughput Under Different Offered Traffic Load. 63
3.8 Pseudo-code of a Predictive Admission Control Scheme. 64
3.9 Throughput of the Adaptive DBAS Framework versus the Predictive CAC [C=200]. 66
3.10 Forced Connections of the Adaptive DBAS Framework versus the Predictive CAC [C=200]. 66
3.11 Completed Connections of the Adaptive DBAS Framework versus the Predictive CAC [C=200]. 67
3.12 Blocked Connections of the Adaptive DBAS Framework versus the Predictive CAC [C=200]. 67

3.13 Effective Throughput (per channel) of the DBAS Framework versus a Priority Based Scheduler. 69

4.1 Example: Preemptive versus Non-Preemptive Scheduling 71
4.2 Integer Linear Program Formulation 77
4.3 A Scenario with Seven Connection Requests 78
4.4 Pseudo-code for Function P2-CPS. 81
4.5 A View of the Solution Computed by Function P2-CPS as a Composition of Multiple Sub-schedules 82
4.6 Three Instances of Requests that can be Served Using two Channels 83
4.7 Pseudo-code of the Time-slotted Predictive CAC Scheme 86
4.8 TS Predictive with and without CAC scheme 87
4.9 Effective Throughput 88
4.10 Forced Terminations Percentage 89
4.11 Completed Connections. 90
4.12 Blocked Connections. 91

5.1 Ring-based Mobility Model 99
5.2 Two Way Symmetric Mobility Diagram 103
5.3 Two-way Symmetric Overload Calculation Diagram 107
5.4 Augmentation Vector Scenario with 4 Rings 108
5.5 Pseudo Code for Overload Calculation Function 110
5.6 Scheduling Algorithm Description 112
5.7 The Number-Based CAC Algorithm 113
5.8 The Power-Based CAC Algorithm 114
5.9 Example of 3 Requests to Illustrate the Delay-based Scheme 115
5.10 The Delay-Based CAC Algorithm Description 116
5.11 Effective Throughput 118
5.12 Forced Terminated Connections 119
5.13 Completed Connections 120
5.14 Blocked Connections 121

List of Symbols

Symbols	*Definition*
BAA	Bandwidth Allocation Algorithm
BER	Bit Error Rate
CAC	Call Admission Control
CDMA	Code Division Multiple Access
GGSN	Gateway GPRS Support Node
GSM	Global System for Mobile Communication
GPRS	General Packet Radio Services
GPS	Generalized Processor Sharing
HLR	Home Location Register
IP	Internet Protocol
KBps	Kilo Byte per Second
Kbps	Kilo Bit per Second
LRD	Long Range Dependence
MAI	Multi-user Access Interference
MIT-2000	International Mobile Telecommunication-2000
MMPP	Markov Modulated Poisson Process
MPEG	Moving Picture Experts Group
PN	Pseudonoise Sequence
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RNC	Radio Network Controller
RM	Resource Management
SGSN	Serving GPRS Support Node
SINR	Signal to Interference and Noise Ratio
SNR	Signal to Noise Ratio
TDMA	Time Division Multiple Access
UDP	User Datagram Protocol
UE	User Equipment
UMTS	Universal Mobile Telecommunication System
UTRAN	UMTS Terrestrial Radio Access Network
WCDMA	Wide CDMA
WLAN	Wireless Area Network

Chapter 1

Introduction

Recent advances in the third generation (3G), and newer generation (NG), cellular networks have shown that such networks can support mobile users with high data rates through both packet switched connections, and circuit switched connections. Such advances enable network service providers to extend the current Internet multimedia streaming services to mobile users.

Provisioning Quality of Service (QoS) for multimedia streaming traffic is currently viewed as critical to the profitable deployment of such networks. Nevertheless, providing the required QoS measures to such traffic faces a number of challenges due to the delay sensitive aspects of the multimedia traffic, and the scarce wireless networks resources.

In this chapter we identify and briefly review some basic concepts required to approach the above class of QoS provisioning problems for delay sensitive traffic in the challenging environment of cellular networks. In particular, Sections 1.2 to 1.8 provide suitable background information and discussions on the following:

1. background on some existing cellular wireless networks,
2. issues on handoff, and user mobility,
3. issues on audio and video traffic modeling, and
4. background on multimedia encoding techniques and multimedia streaming.

Chapter 2 utilizes the background information presented here to review some specific network design techniques for provisioning QoS in wireless networks.

1.1 Introduction

Recent advances in wireless personal communication systems have led to increasing interest in the engineering of various types of wireless networks, for example, cellular networks for wide area coverage, mobile ad-hoc networks (MANETs) for quick deployment, and wireless enabled sensor networks.

Supporting node and user mobility, and achieving better integration of such networks with the global Internet through the support of high data rate packet services have emerged as important design aspects of such networks.

For cellular networks, achieving the above goals has led to a vast amount of research at different layers of the networking protocol stack. The work done so far spans the following areas: managing the wireless network resources during user handoff for voice calls, development of protocols to support mobility, and development of resource management schemes to support heterogeneous mixes of voice and data traffic.

In this thesis, we consider problems arising in delivering delay bounded traffic to mobile users of cellular networks. Such traffic is important to support (for example, for delivering multimedia streaming services), yet challenging to manage, since it typically requires relatively high data rates that consume significant wireless system resources. Moreover, users expect uninterrupted operation while roaming within the coverage area.

To date, the work proposed by different researchers for provisioning quality of service (QoS) in cellular networks has emphasized the need to develop suitably sophisticated call admission control (CAC) mechanisms, and packet scheduling algorithms as the main tools to manage the scarce wireless resources.

This chapter aims at summarizing some of the important issues required to investigate the above directions further. Section 1.2 provides an overview of cellular networks. Section 1.3 presents some key concepts of the wireless layer. Section 1.4

presents resource management algorithms. Section 1.5 presents some issues related to handoff and mobility models. Section 1.6 discusses multimedia traffic modeling. Section 1.7 review some issues related to multimedia encoding. Finally, Section 1.8. gives an example of a multimedia streaming session over UMTS network. Finally, Section 1.9 summarizes the chapter and provides a general overview of the thesis structure.

1.2 Background on Some Cellular Networks Architectures

Cellular wireless networks are the primary means of extending wireless connectivity (with data rates currently about 2 Mbps in some cellular standards) to mobile users over wide geographical areas (e.g., cities and highways). Their services are particularly important in areas where coverage by wireless local area networks (WLANs) does not exist. In this thesis, we consider developing resource management algorithms for two main types of cellular networks: networks that employ time division multiple access (TDMA) for multi-user access, and networks that employ code division multiple access (CDMA) for multi-user access. In the following part we briefly mention some of the important architectures and standards for networks of each type.

1. GPRS (General Packet Radio Services) [62] is viewed as a 2.5G architecture that employs TDMA to provide both voice, and data to mobile users. GPRS provides an evolution path for supporting data transmission using the voice-based GSM (Global System for Mobile communication) networks [58].

2. UMTS (Universal Mobile Tele-communication System) [62] is viewed as a 3G architecture that employs CDMA to provide both voice, and data to mobile users. UMTS employs CDMA air interface (e.g., the WCDMA (Wide CDMA) standard [30], or the cdma2000 standard [24], both standards are successful proposals that satisfy the IMT-2000 requirements [36]) to provide mobile users with the following data rates: up to 2 Mbps for stationary users, and up to 144 Kbps for moving users.

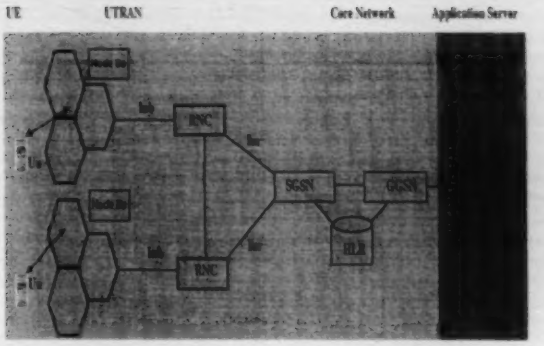

Figure 1.1: *UMTS Network Architecture*

Figure 1.1 presents a general UMTS network architecture. End to end delivery of multimedia services over UMTS networks architecture consists of a user (User equipment UE) requesting a multimedia clip from a server. Users are connected to the base station (Node B). Each Node B is connected to a radio network controller (RNC). The UMTS terrestrial radio access network (UTRAN) is then connected to the UMTS packet switching core network via serving GPRS support node (SGSN). The core network uses the gateway GPRS support node (GGSN) to connect to the multimedia server. The core network is responsible of switching, routing and transmitting users traffic. Home location register (HLR) is another component of the core network.

UMTS supports four different types of services: *conversational*, *streaming*, *interactive*, and *background*. The conversational class is designed for delay sensitive, real time applications, such as videoconferencing. The streaming class is designed for non real time applications, such as video on demand. The other two types are used for non-real time applications. In this thesis, we are concerned only with the streaming class that supports one-way real time applications.

Finally, we note that the cell capacity in CDMA-based architectures is limited by multi-user access interference (MAI) [66], as described in Section 1.3.

1.3 Power and Interference Aspects in Wireless Networks

Certain parts of this thesis discus serving delay bounded traffic in CDMA networks. To ensure, effective admission control, the devised scheme deals with certain physical layer constraints. In particular, the resource management schemes are designed to secure a satisfactory level of *signal to interference plus noise ratio* (SINR) to each active mobile user. In this regard, it is important to consider two issues:

1. a large scale path loss model for signals transmitted from the base station to a mobile user, and

2. the multi-user access interference (MAI) that plays an important role in determining the available bandwidth in CDMA cellular networks.

We review these aspects below.

A. Large Scale Path Loss. The amount of power received by the user is smaller than the actual amount sent by the base station, due to path losses. Path loss is classified into *large* scale path loss, and *small* scale path loss [65]. The path loss model used here is known as the *log-normal shadow fading* model for large scale path loss. According to the log-normal shadow fading model, the path loss between two points that are separated by distance d, denoted $PL(d)$, is a normally distributed random variable, that is

$$PL(d) = \overline{PL}(d) + X_\alpha \quad (dB\ units) \tag{1.1}$$

where,

1. $\overline{PL}(d)$ is the average path loss, and it is computed as follows:

$$\overline{PL}(d) = \overline{PL}(d_0) + 10 \times n \times log(\frac{d}{d_0}). \tag{1.2}$$

2. $\overline{PL}(d_0)$ is the path loss at a close-in reference distance d_0. $\overline{PL}(d_0)$ is determined based on either a close-in measurement, or a free space model between the mobile user and the base station [65].

3. n is an exponent whose value is determined by the environment. For free space, $n = 2$. Larger n values are used to model environments with obstructions [65].

4. X_α is normally distributed random variable with zero mean (in dB), and a standard deviation equals to α (also in dB).

Example 1:

To predict a path loss for a user that is at a distance $d = 500$ meters away from a base station, assume that at $d_0 = 711$ meters, the measured average path loss $\overline{PL}(d_0) = 142$ dB. In addition, assume that $n = 4$, and $X_\alpha = 6$ dB. Using Equations 1.1, and 1.2, the path loss at distance $d = 500$ meters is about 141.8 dB. For another user that is 800 meters away from the base station, and using the same parameters, the path loss between this latter user and the base station is about 150 dB. □

B. Multi-user Access Interference in CDMA Systems. The goal of the multi-user access scheme is to divide a shared medium among the users. CDMA allows many users to transmit at the same time. As described in [38], the base station transmits to all active users at the same time using the same frequency band, which means that each user receives the data sent to all other users and considers it as interference. In CDMA, each user, say user i, is assigned a primary pseudo-noise (PN) code C_i^{PN} that the user utilizes to extract his own data.

Each base station has a limited amount of available power, denoted P_{total}, for downlink data transmission. The base station uses such power for transmitting data, and control signals (e.g., pilot, power control and synchronization signals) to mobile users. The power used by the base station to transmit data to a particular user should be adjusted so as to obtain (or exceed) a certain required SNR at the receiver's end. The SNR experienced by a receiver depends on the received power from the serving base station, as well as the received power from interfering base stations in neighboring cells, as specified by Equation 1.3 below.

In Equation 1.3 [38] the left hand side is assumed to be a minimum target SNR (also referred to as energy per bit to interference ratio) that must be achieved for a

typical user, denoted by the index i, to correctly receive data. The right hand side is a ratio of the received signal power (per bit) from user i serving base station (the numerator) to the total interference power from transmission of the serving base station to other mobiles in the same cell (the $\alpha.I_{intracell}$ term), the interference from neighboring base stations (the $I_{intercell}$ term), and the receiver's white Gaussian noise (the $\eta_0 * W$ term):

$$SNR_i \leq \frac{SF_i \times P_{r,i}}{\alpha.I_{intracell} + I_{intercell} + (\eta_0 \times W)} \tag{1.3}$$

where,

1. SF_i is the spreading factor of user i defined as (W/R_i), where W is the chip rate used by the network (described below), and R_i is the rate of the received data,

2. $P_{r,i}$ is the received power from the serving base station. $P_{r,i}$ is estimated as the following:

$$P_{r,i} = \frac{base\ station\ transmitted\ power\ to\ user\ i\ (P_{t,i})}{path\ loss\ between\ base\ station\ and\ user\ i\ (PL_i)} \tag{1.4}$$

3. α is an orthogonality factor (e.g., between 0.1 and 0.4), that compensates for loss of orthogonality due to multipath [38],

4. $I_{intracell}$ is the interference power from the serving BS inside the cell,

5. $I_{intercell}$ is the interference from neighboring BS, and

6. η_0 is the noise density.

7: W is the chip rate of the system (e.g., 4.096 Mega chips per second (Mcps) for the WCDMA standard),

8. R_i is user i data rate,

9. PL_i is the path loss from the serving BS to user i,

For a particular user (denoted i above), Equation 1.5 estimates the interference caused by the transmission of user i serving base station to mobiles in the same cell:

$$I_{intracell} = \sum_{j \neq i} \frac{P_{t,j}}{PL_i} \tag{1.5}$$

And, Equation 1.6 estimates the interference power from base stations in the neighboring cells on user i. The major part of the intercell interference comes from the first two tiers: 6 cells in the first tier, and 12 cells in the second tier [38]. Figure 1.2 illustrates the 2-tiers model layout.

$$I_{intercell} = \sum_{c \neq bs} \frac{P_{t_{c,i}}}{PL_{c,i}} \tag{1.6}$$

where,

1. bs is the base station that serves the target mobile user (denoted i) under consideration, and
2. $PL_{c,i}$ is the path loss between base station c (a base station in the first, or the second tier of the neighboring cells), and user i.

The following numerical example illustrates the use of the above relations to estimate the required base station transmission power to support transmission at certain data rates to users.

Example 2:

In this example, we consider serving two users at a data rate of R bps in a CDMA network. By Equation 1.3, the minimum amount of transmission power to each user ($P_{t,i}$ for $i = 1, 2$) should satisfy the following equations for $i = 1, 2$:

$$SNR_i \leq \frac{SF_i \times \frac{P_{t,i}}{PL_i}}{\alpha . \sum_{j \neq i} \frac{P_{t,j}}{PL_i} + \sum_{c \neq bs} \frac{P_{t_{c,i}}}{PL_{c,i}} + (\eta_0 \times W)}.$$

Rewriting the above two equations in the two unknowns $P_{t,1}$, and $P_{t,2}$, we get:

$$\begin{bmatrix} SF_1 & -\alpha(SNR_1) \\ -\alpha(SNR_2) & SF_2 \end{bmatrix} \begin{bmatrix} P_{t,1} \\ P_{t,2} \end{bmatrix} = \begin{bmatrix} SNR_1((\eta_0 \times W \times PL_1) + \sum_{c \neq bs} \frac{P_{t_{c,i}}}{PL_{c,i}}) \\ SNR_2((\eta_0 \times W \times PL_2) + \sum_{c \neq bs} \frac{P_{t_{c,i}}}{PL_{c,i}}) \end{bmatrix}$$

In this example, for simplicity, we ignore the inter-cell interference power $I_{intercell}$. We now use the following values:

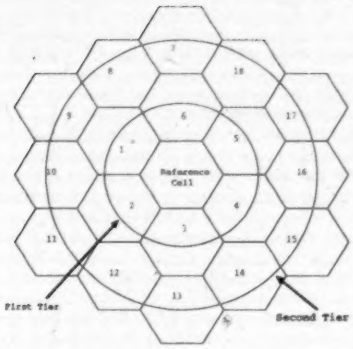

Figure 1.2: *Two Tier Cell Structure Model [77]*

- $W = 4.096$ Mcps, and data rate $R = 128$ kbps (that is, $SF = W/R = 32$).
- α (orthogonality factor)= 0.2, and both users require the same SNR value, $SNR = 5.0118$ (7 dB), thus $\alpha SNR = 1.0023$.
- η_0(noise density) is $3.98 \times 10^{-18}\ mWatt/Hz$.
- Assuming $PL_1 = 1.543 \times 10^{-14}$ $(141.88\, dB)$, and $PL_2 = 1.0113 \times 10^{-15}$ $(150\, dB)$ (as in Example 1).

we get

$$\begin{bmatrix} 32 & -1.0023 \\ -1.0023 & 32 \end{bmatrix} \begin{bmatrix} P_{t,1} \\ P_{t,2} \end{bmatrix} = \begin{bmatrix} 82627.0 \\ 12606.9 \end{bmatrix}$$

Solving the above system of equations, results in $P_{t,1} = 486.7$ mWatts, and $P_{t,2} = 2659.9$ mWatts. □

We note that at any instant the sum of transmission power assigned by the base station to all its served users is limited by the maximum transmission power available at the base station denoted P_{total}. Equation 1.7 expresses this constraint:

$$P_{total} \geq \sum_{i=1}^{N} P_{t,i} \tag{1.7}$$

Table 1.1 summarizes the symbols used in this section.

Table 1.1: Mathematical Symbols Summary Table

Symbols	*Definition*
P_{total}	Maximum power available for transmitting data at the base station
$P_{t,i}$	Data transmission power used by the serving base station to user i
$P_{r,i}$	Power received by user i from the base station
SNR	Signal to noise ratio
SF_i	Spreading factor for user i
W	Chip rate
PL_i	Path loss between the serving base station and user i
α	Orthogonality factor
$I_{intracell}$	Interference power inside the cell
$I_{intercell}$	Interference power from the neighbor cells
η_0	Noise density
d_i	Distance between user i and the base station
d_0	A reference distance used in the log-normal shadow fading model

1.4 Resource Management in Cellular Networks

Resource Management (RM) algorithms such as power control, handover control, call admission control (CAC), and packet scheduling, are used to control radio resources in cellular networks among the active users.

CAC and packet scheduling algorithms are two key components of the RM suite that aim at maximizing the network service provider profit while satisfying users' QoS requirements. Call admission control algorithms decide whether to accept or reject a new connection. The main goal of CAC algorithms is to maximize the system throughput. On the other hand, packet scheduling algorithms distribute the resources available at the base station among the user packets. The main goal of packet scheduling algorithms is to maximize the effective throughput. Maximizing both the system throughput and effective throughput leads to an increase in the network service provider profit. Increasing the effective throughput guarantees users' satisfaction.

In Chapter 2 we present some of the work done in the literature for both CAC and packet scheduling.

1.5 Issues with User Mobility

A challenging aspect of designing effective resource management algorithms for wireless/mobile networks is the presence of user mobility while a user is actively communicating with a base station. We note that the exact effect of different types of mobility (for example, mobility within a single cell, between two cells that are controlled by the same switching center, or between two cells that are controlled by different switching centers) depends on the multiple access method (e.g., TDMA or CDMA) used by the network architecture. Below we present some important mobility aspects by discussing issues in the following areas: (a) handoff, and (b) mobility modeling.

A. Handoff Issues

Handoff is the process of moving from one cell to a neighboring cell. To avoid service interruption, this requires a seamless change in the serving base station. The main problem with handoff is that it is possible that the new base station doesn't have enough resources to keep the new handoff call connection.

A distinction between a new handoff call and a new originated call is made at the base station. Dropping probability is the probability that the base station will reject a handoff call. Blocking probability, on the other hand, is the probability that the base station will reject a new originated call. The balance between dropping and blocking probabilities is a major issue in designing CAC algorithms. For voice calls, it is more important to continue an ongoing connection (reducing the dropping probability) than to accept a new connection. Therefore, minimizing the dropping probability is an important objective of most of the proposed CAC schemes. CAC schemes using this technique are called priority based schemes. Priority based schemes are generally classified into two main schemes [32]:

1. Admission control schemes based on defining *guard* Channels (also called *trunk reservation schemes*): In [63], the authors propose a guard channel scheme that reserves a fixed number of channels at the base station for handoff. The work of [64] presents an enhanced method called the *fraction guard*

channel scheme. In [50], the authors propose the concept of *shadow cluster* that reserves handoff channels based on the estimated resources, taking into account current users' mobility patterns. Another mobility based CAC algorithm is proposed by Hou et al. [32].

2. Admission control schemes based on using priority queueing: both handoff and new calls are queued [31, 72, 47]. No resources are being reserved for handoff calls in these schemes. Handoff calls queues are given a higher priority over new calls queues.

The above methods deal with handoff traffic of circuit switched voice calls. For packet switched services, all packets are subject to delays, and network resources are allocated to satisfy the QoS parameters of each traffic stream.

B. Mobility Modeling

Mobility in wireless/mobile networks can be classified into (see, for example, the classification presented in Section 8.2 of [37]):

- *Individual* mobility models: in an individual mobility model, each user movement is specified independent of other users. Random way model [9] is an example of an individual mobility model that assumes a user can move in any one of a set of directions (or to a set of new locations) with equal probability. Further information about this model is presented next.

- *Aggregate* mobility models: in an aggregate mobility model, a relationship among a group of users and their movement patterns (both inside the cell and between cells) exists. Fluid model [8] is an example of an aggregate model that averages the mobility patterns of all the users.

Another classification for mobility models is based on the prior knowledge and information that the model assumes about the users, (see Section 8.2 of [37]). In this regard, models can be classified as prediction based and non-prediction based. Random movement model considers no prior knowledge of the user movement and the user moves with equal/certain probability to each direction and with random

speed. The shortest-distance model is an example of a prediction based model, which assumes certain knowledge about the users movement behavior such as their source and destination locations.

1. **The Random Way Model.**

   One of the most used mobility models in mobile network is the random way mobility model. In this model, users are represented by their initial location (x, y) in a region. The initial locations are chosen independently and uniformly in the region. Users stay in their location for a random amount of time (pause time). When the user decides to move, he picks a new location (nx, ny) independently and uniformly in the region. This model also chooses a moving speed uniformly on interval (v_0, v_1). Once the user reaches its destination it stops there for some random time (pause time). It is possible that the user decides to move again as soon as it reaches its destination. Users keep repeating the same pattern over and over.

2. **A Model with User Residence and Movement Distributions.**

   In this section we present a mobility model that has been shown to be useful in the literature [46, 51]. Our work in Chapter 5 follows a similar approach. The model basic assumptions are:

   (a) Users move from one region to another. A cell is an example of a region. Users are assumed to be randomly assigned to regions: a user may stay in the same region, or decide to move to a neighboring region.

   (b) The time user i spends in a region j is called *dwelling time* T_{ij} which is randomly generated.

   Assume that user i is in region C and the user starts a call, the call duration is T_i. The dwelling time for user i in region C is T_{ic}. In case the dwelling time is less than the call time, a handoff from the current region to a new region occurs.

   (c) Some models allow only one handoff during the call duration. Other

models allows more than one handoff. In case of handoff, the user decides to move to neighbor regions with probability P_h.

The model opens the door for computing a number of interesting probabilities. For example, in [46] the authors have utilized the above model to compute the probability of having k users in a cell C by the end of prediction interval of T_e seconds (in the future), as described below.

The Work of [46]:

The following is an example of the above model. Assume we have three neighboring regions ($C, L, and\ R$). C is the middle region, L is the left neighboring region of C, and R is its right neighboring region. The number of users in each region is denoted by: n, l, and r. Let i be a user in region C. The probability that i stays in the same region is P_r, and the probability he moves to one of the neighboring regions is P_h. The probability of moving to region L is $P_h/2$ and the probability of moving to region R is $P_h/2$ as well.

As mentioned before, the model predicts the number of users in each region at the end of the estimation time T_e. The number of users at any region by the end of the estimation time is equal to the number of users at the start time t_0, plus the number of users moved to the region during time T_e, minus the number of users moved away from the region during the same time T_e. Let the number of users in region C at time t_0 be n. The probability of i_n users remaining in region C by the end of estimation time $t_0 + T_e$ has a binomial distribution ($B(i_n; n, P_r)$). Also, the probability of i_l users leaving region C to the left region follows a binomial distribution ($B(i_l; l, P_h/2)$) and the probability of i_r users leaving region C to the right region follows a binomial distribution ($B(i_r; r, P_h/2)$). The Binomial distribution is defined as the following:

$$B(i; n, p) = \left(\begin{array}{c} n \\ i \end{array} \right) p^i (1-p)^{n-i} \tag{1.8}$$

The probability of having k users in region C by the end of the estimation

time ($t_0 + T_e$) is denoted by $P_{t_0+T_e}(k)$ and is evaluated using the convolution sum of three binomial distributions: $(B(i_n; n, P_c))$, $(B(i_l; l, P_h/2))$ and $(B(i_r; r, P_h/2))$, where $k = i_n + i_l + i_r$.

It is known that the summation of binomial distributions can be approximated by Gaussian distribution [46, 33]. Taking this into account we can conclude that $P_{t_0+T_e}(k)$ approximate Gaussian distribution with mean $nP_r + (l+r)\frac{P_h}{2}$ and standard deviation $\sqrt{nP_r(1-P_r) + (l+r)\frac{P_h}{2}(1-\frac{P_h}{2})}$ as follows:

$$P_{t_0+T_e}(k) \simeq N\Big(nP_r + (l+r)\frac{P_h}{2}, \sqrt{nP_r(1-P_r) + (l+r)\frac{P_h}{2}(1-\frac{P_h}{2})}\Big) \tag{1.9}$$

1.6 Traffic Modeling

Traffic modeling is an important aspect of any performance analysis study. For our work here, the most important types of delay sensitive traffic are audio and video traffic. Currently, many analytical models of such traffic, with various degrees of accuracy have been devised. We note, however, that due to the complexity of solving many wireless resource management problems, many analytical studies make simplifying assumptions about the input traffic. In particular, many studies describe an audio or video stream as a connection that

1. has a duration drawn from a certain distribution (usually the exponential distribution), and

2. is best served at a certain data rate. For example, the maximum data rate for audio traffic is 4.8 Kbps [40].

In Chapters 3, 4 and 5, we use the above model to generate user traffic. Examples of other models include:

1. Poisson and Markov models. These models are used frequently in the literature due to their simplicity.

2. Models based on the concepts of self-similar, and long range dependent (LRD) traffic. These models are predicted to play more significant roles in the data networks performance evaluation and modeling (see Chapter 6 of [37]).

3. Two state models (ON/OFF models). Such type of models are used to model audio traffic. The first state represents the active (talk spurts) interval and the second state represents the silence interval. The length of each state is assumed to be exponentially distributed.

4. Fluid source model. Usually used to model audio traffic.

5. Markov modulated Poisson process (MMPP) models. Used to model both audio and video traffic. A MMPP uses Markov chain with N states to modulate a Poisson process. State i generates traffic from Poisson process with parameter λ_i. λ_i is state-dependent and is different for each state.

1.7 Issues with Multimedia Encoding Techniques

The design of wireless data networks faces the challenges imposed by the limited availability of sufficient bandwidth to enable mobile users to enjoy high quality audio and video components. To help in relieving such limitations, research work has considered ways to identify possible opportunities of exploiting the intrinsic properties of the used audio and video coding techniques to make efficient use of network resources. In this context, the following opportunities have been discussed in the literature.

1. When the wireless network experiences an overload condition (e.g., due to the incoming handoff traffic, or users moving away from base station and thus requiring more transmission power), the resource management algorithms may decide to degrade the bandwidth of a subset of users. If such users are being served by a multimedia server capable of adjusting the traffic encoding parameters so as to accommodate requests of the wireless network then the network may be able to achieve better performance by minimizing the number of forced connection terminations.

2. Some encoding techniques (e.g., transform coding) allow some video frames to be dropped when congestion arises in the network, and later compensate for this receiver by a suitable interpolation mechanism. In such cases, the incoming UDP packets to a base station may carry flags reflecting a packet's dropping priority.

In Chapter 2, we review a CAC that utilizes the first type of opportunities to manage the network resources [46]. Below we mention more details on multimedia encoding that have helped in providing the opportunities in the two points mentioned above.

A. Codec techniques. A codec refers to the hardware or software used to convert analog data to digital stream. It involves analog to digital and digital to analog conversion and/or compression and decompression. In [74], the authors presents a classification of codec techniques into the following classes: *transform* coding, *sub-band* coding vector quantization, and *region* encoding. MPEG and H.261 are examples of the transform coding technique.

B. The MPEG standards. MPEG is one of the most successful codec formats. The key method of achieving a high compression rate is to store the changes from one frame to another instead of storing the whole frame. The three major standards for MPEG are

1. MPEG-1: Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mb/s.

2. MPEG-2: Generic coding of moving pictures and associated audio.

3. MPEG-4: Provisioning of a very high quality audio and video streams.

There are three types of frames in MPEG standard: I (Intra-frame), P (Predicted) and B (Bi-directional) frames. The I type encodes still image and may be viewed as a starting frame. The P frames are the predicted frames from either I or P frames, while B frames are generated from two I or P frames such that the B frame will be

in between these two frames (i.e., one past frame and one future frame are used to generate B frame). A possible sequence of MPEG frames is:

$$(I_1 B_1 B_2 P_1 B_3 B_4 P_2 B_5 B_6 P_3 B_7 B_8 I_2 B_9 \ldots)$$

Where for example, B_1 and B_2 are generated from I_1, and P_1. P_1 is a predicted frame based on I_1.

1.8 Multimedia Streaming

The subject of providing QoS for multimedia traffic over wireless networks is an ongoing research area. Most of the work done in this area deals with multimedia traffic in terms of its sensitivity to delay and jitter [25, 70, 20]. Some of the work is done in the area of heterogeneous traffic [84, 82]. The integration of codec characteristics has been studied for wireless networks in [79].

In general, network applications can be classified into three categories:

1. Human to human (e.g., videoconferencing and voice telephony).

2. Human to computer (e.g., voice and video playback services). Applications under this classification can be categorized into three main types as the follows: (a) Audio only applications, (b) Audio associated with still pictures (Navigation systems is an example for such application), and (c) Video playback.

3. Computer to computer (e.g., multicast feed, news feed, batch processing and database synchronization).

Multimedia streaming is the process of transporting one or several media streams over the network to mobile users. Human to computer applications is an example of multimedia streaming. The main two advantages of multimedia streaming are [56]:

1. Streaming multimedia does not require large terminal buffer, and

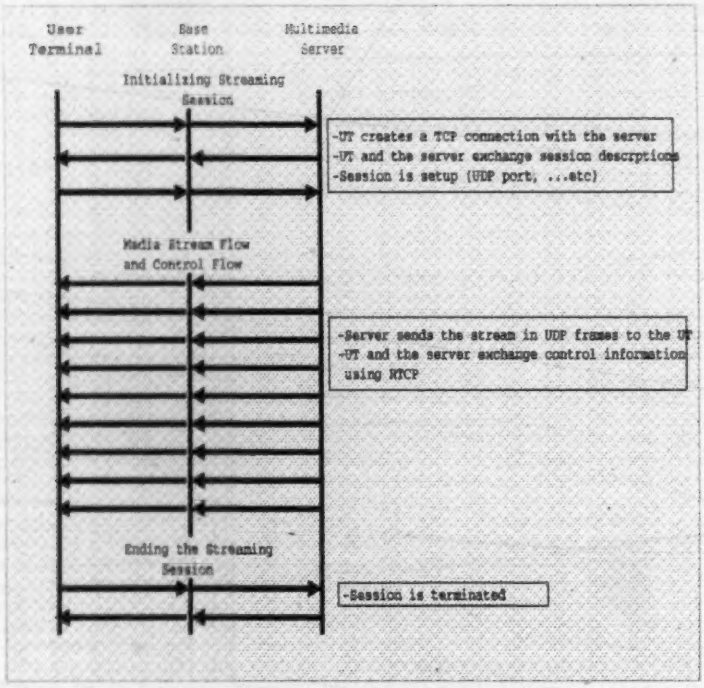
User Terminal
Base Station
Multimedia Server
Initializing Streaming Session
-UT creates a TCP connection with the server
-UT and the server exchange session descriptions
-Session is setup (UDP port, ...etc)
Media Stream Flow and Control Flow
-Server sends the stream in UDP frames to the UT
-UT and the server exchange control information using RTCP
Ending the Streaming Session
-Session is terminated

***Figure 1.3:** Example of Multimedia Streaming Flow*

2. Streaming multimedia is suitable for transmitting live events to users as they happen.

A. Real Time Protocol for Multimedia

Real time streaming protocol (RTSP) [67] is an example of the most commonly used streaming protocols. RTSP is a control protocol for streaming multimedia to the user(s) from a multimedia server, the main function of the RTSP is to initiate a multimedia stream connection between the user and the server, it is also used to control the connection. RTSP may use realtime transport protocol (RTP) to transmit the streamed data. Figure 1.3 depicts the data flow between the user and the server.

In a successful scenario, the following basic events occur: (1) a client requests a media clip from the server, (2) a TCP connection is established between the client and the server, (3) the client sends a description message containing useful information to playback the clip, (4) the server acknowledges the message, (5) the client sends a session's setup message that needed to be confirmed by the server, (6) two UDP connections are created to carry RTP and RTCP traffics, respectively.

B. Delay Jitter

The multimedia flow is generated at fixed rates and is expected to reach the end terminal at fixed rate as well. The variations in queueing delays experienced by individual packets that may influence the quality of the transmitted media flow. This variation is refereed to as jitter. Many proposals have been introduced in the literature to deal with the effect of jitter [16, 6]. In order to accommodate the jitter's influence, the end terminal uses a play-out buffer. This buffer delays the incoming packets to smooth the stream playback.

In [6] the authors introduced a network model (jitter graph) to capture the characteristics of different service disciplines (i.e., bandwidth, jitter and reliability). The jitter graph is then employed to design a traffic regulator called initial delay regulator (IDR) that aims at eliminating jitter.

1.9 Summary

In this chapter, we have presented some background information related to the problems formulated in the next chapters. The chapter presents a brief introduction to wireless networks architectures, user mobility modeling, traffic modeling, and multimedia streaming. Several aspects of the above mentioned areas are used to formulate the works in Chapters 3, 4 and 5. The rest of the thesis is organized as the following:

Chapter 2 provides an overview of some of the related research work in the area. Chapter 3 formulates a call admission control problem for delay bounded traffic, the devised framework utilizes a non-preemptive scheduling mechanism. Similar framework that utilizes a preemptive scheduler is considered in Chapter 4. For both Chapters 3 and 4, the proposed schemes are developed for networks with fixed number of dedicated channels. In Chapter 5, a call admission control scheme is proposed for soft capacity systems. The thesis is concluded in Chapter 6 with summary, conclusion remarks and future work directions.

Chapter 2

Literature Review

Currently, there is a vast number of published research work in the general area of designing resource management algorithms for providing quality of service guarantees to mobile wireless users. The early well-established results in this regard have focused on the design of call admission control (CAC) algorithms for minimizing call blocking probability and forced termination probability for handoff voice calls. Section 2.1 below discusses some results related to this direction.

Subsequent work in the area has considered a variety of scenarios such as, support of heterogeneous mixes of voice, and data traffic of certain types. Section 2.2 provides a broad overview of some work relevant to the area of provisioning QoS over wireless/mobile networks. Section 2.3 presents an overview of some scheduling problems. In Sections 2.4 to 2.6, we identify and discuss some research results that are related to the thesis; the contexts of the surveyed results are as follows: Section 2.4 deals with the work in [46] where the authors propose call admission control and adaptive bandwidth allocation schemes for delivering multimedia traffic to mobile users in cellular networks where a fixed number of channels are assumed to be allocated for serving a target traffic. The data rate can be decreased (when overload conditions arise), or increased (when some connections terminate) dynamically. The devised method works by predicting the state of a cell (the number of active users, including handoff users) after some prediction time

interval in the future.

Section 2.5 deals with the work in [39] where the authors consider the performance of a variety of scheduling algorithms (that determine which connections to serve, when, and at what data rate) given that the traffic volume in each connection is known in advance (e.g., as in downloading web objects from a server).

Section 2.6 deals with the work in [80] where the authors consider the design of a scheduling mechanism that uses a generalized processor sharing (GPS) framework for allocating data rates to a heterogeneous mix of traffic in a CDMA soft capacity environment.

2.1 Admission Control for Voice Traffic

One of the major areas covered in the literature is the provisioning of voice services over networks with fixed capacity. The following summarizes some related work.

1. Posner and Guerin's [63] work is an early proposal for using the guard channel (trunk reservation policy) technique to handle the handoff problem. They propose that each base station reserves a fixed number of channels for the handoff calls. Several techniques based on the original work of [63] have been later proposed. The main improvements in such proposed technique rely on:

   - dynamically adjusting the number of reserved channels at each base station [59, 50, 32, 1], and
   - supporting different traffic types [22, 51].

2. Ramjee et al. [64] studied the optimality of both the guard channel policy [63], and a fractional guard channel policy introduced in [64]. The authors have examined three problems:

   (a) Minimizing a linear objective function that considers both the new call blocking probability and the handoff call dropping probability: the de-

vised problem solution is shown to be optimal using the guard channel policy.

(b) Minimizing the new call blocking probability with a hard constraint on the handoff call blocking probability: the devised problem solution is shown to be optimal using the fractional guard channel policy.

(c) Minimizing the number of channels required to satisfy hard constraints on the handoff and new call blocking probabilities: the devised problem solution is shown to be optimal using the fractional guard channel policy.

3. Epstein and Schwartz [22] proposed three call admission control algorithms for wireless networks that support different classes of user traffic. The proposed methods are coined the following names:

   (a) Complete sharing (CS): in this method all users are treated equally.

   (b) Complete partitioning (CP): in this method the bandwidth is divided into sub-pools according to users' traffic characteristics.

   (c) Hybrid reservation: this method is a mix between both CS and CP methods.

   Due to the complexity of the problem, the authors limit the number of traffic classes to 2. In [51], Li et al. propose a hybrid cutoff priority scheme for wireless networks that can support N, $N \geq 1$, traffic classes.

4. Naghshineh et al. [59] proposed a distributed CAC algorithm the works by estimating the cell overload probability at the end of a prescribed estimation time, and then deciding whether to accept or reject a new arrived call. The estimation takes into account the number of users in the current cell as well as the number of users in the adjacent cells and the handoff probabilities from adjacent cells to the target cell under analysis. The main advantage of the proposed CAC is its relative simplicity, which makes it attractive for use in real systems.

5. Levine et al. [50] used the concept of *shadow cluster* to predict system status. A *shadow cluster* of a mobile user is defined to be a group of base stations that the mobile user is likely to move to during a single voice call duration. The base stations in a cluster are assumed to share this information and adjust the number of reserved guard channels accordingly. The accuracy of the proposed algorithm depends on good prediction of user movement direction, which makes it more suitable for direction oriented environments such as highways. A similar concept that is called *influence curves* has been proposed in [32].

2.2 Admission Control and Scheduling for Mixed Voice and Data Traffic

Subsequent work in the area has combined mixed traffic of voice and data. Examples of such work include the work mentioned below. We recall that the available bandwidth in networks that utilizes the CDMA multi-user access is affected by the multi-user interference (MAI), and hence special attention is given for allocating the transmission power for each user.:

1. CAC algorithms for CDMA networks [12, 2, 81, 75, 69].

2. Soh and Kim [69] propose a dynamic bandwidth reservation scheme which tries to predict user mobility, and accordingly adjust the amount of power reserved for handoff calls. The authors use road topology to achieve acceptable prediction.

3. Wang et al. [75] propose two resource management schemes for downlink transmission in CDMA network. The proposed schemes are:

   (a) Guard capacity adaptation based on dropping (GAD) scheme: In the scheme the cell reserves some of its capacity for the handoff calls (guard capacity). The cell dynamically adjusts the guard capacity by monitoring handoff to maintain dropping rates below a target level.

(b) Guard capacity adaptation based on prediction and dropping (GAPD) scheme: In addition to monitoring the handoff, the cell in this scheme tries to predict handoff. The cell uses the pilot signal strength for the prediction.

4. Xiao et al. [79] propose an optimal call admission control algorithm that supports multiclass adaptive multimedia traffic. The authors modeled the problem as a semi-Markovian decision problem, and use an interior-point linear programming technique to solve the resulting problem.

5. Yu et al. [81] propose a QoS scheme for multiclass adaptive multimedia traffic, the authors use a similar model to that in [79]. The novelty in this scheme is that the proposed algorithm takes into account the status of the neighboring cells along with the status of the home cell. In order to overcome the dimensionality problem that arises when using linear programming to solve the model, the authors use the *average reward reinforcement learning* technique to maximize the system profit, taking into account QoS constraints.

6. Chen et al. [12] extended the concept of guard channels for TDMA networks in addition to the load curve to introduce the *interference guard margin* (IGM) concept for CDMA networks. In this scheme the base station dynamically reserves a certain number of resources for high priority call. The load curve is used to predict the interference level resulting from a given value of the offered traffic.

7. In [14] the authors propose dynamic resource allocation scheme for forward links of the 1x EV-DV standard [57] of CDMA cellular networks. The proposed scheme integrates the QoS requirements (e.g., data rate and delay) at the link layer with the system parameters at the physical layer. The integrated model is an extension of the effective capacity model of [76]. The main approach taken is to devise an effective mechanism to translate QoS requirements into what is called a *service curve* [15]. A service curve aims at specifying the minimum amount of data that has to be transfered to the user

during a network defined time slot period to satisfy the user QoS requirements.

8. In [49], the authors propose an optimal resource allocation scheme that supports dynamic sharing of bandwidth between high priority voice calls, and low priority data transmissions. The proposed scheme is then modeled using Markovian model to obtain and optimal allocation for voice and data calls. We note that this model does not consider user delays.

2.3 Scheduling Algorithms

Scheduling is one of the key mechanisms for QoS provisioning. Scheduling problems related to QoS provisioning that has been considered in the literature can be classified into two broad classes of problems: *task* scheduling problems and *packet* scheduling problems. In this thesis, we make frequent references to both classes. Below, we briefly introduce and list some known results for each class.

2.3.1 Task Scheduling Problems

A typical scheduling problem in this class (see, for example, [4, 26]) specifies the following:

- A set of tasks to be scheduled over a number of processors where the tasks can be scheduled either in a non-preemptive or preemptive way.
- Each task may be associated with a release time, a duration, and/or a due date (or deadline).
- A partial order that specifies precedence (or dependence) between tasks may also be specified.
- An objective function that is required to be optimized by the sought after schedule.

The work of [39] is an example where task scheduling problems arise in managing traffic connections in cellular networks. The following are some known results and observations for this class of problems:

1. Many single processor non-preemptive scheduling problems can be approached as problems on the class of interval graphs [34, 29, 71].

2. For single processor preemptive scheduling problems:

   - The shortest remaining time (SRT) scheduling discipline is known to minimize the average completion time of tasks specified by release time, and task duration.
   - The earliest deadline scheduling discipline is known to minimize the maximum lateness of tasks specified by release time, and task duration.

3. For multi processor preemptive scheduling problems: the McNaughton rule [13] provides an efficient optimal solution for minimizing the maximum makespan (completion time).

2.3.2 Packet Scheduling Problems

A basic scheduling problem in this class (see, for example, [42]) specifies the following:

- a shared link with capacity of C bits/seconds,
- a set of N queues, each queue holds packets from one, or more, traffic streams ready for transmission over the shared link (each queue corresponds to a traffic flow in the model), and
- each queue $i, i = 1, 2, \ldots, N$ is associated with a weight, denoted ϕ_i.

We seek a scheduling discipline that satisfies the following conditions: if we denote by $S_i(\tau, t]$ the amount of flow i traffic served during the interval $(\tau, t]$, then we require that

$$\frac{S_i(\tau,t]}{S_j(\tau,t]} \geq \frac{\phi_i}{\phi_j}, \; j = 1, 2, \ldots, N \tag{2.1}$$

to hold for any flow i that is backlogged throughout the interval $(\tau, t]$.

The Generalized Processor Sharing (GPS) scheduling discipline is an idealized discipline (works at the bit level, not the packet level) that achieves the above goal. In GPS, the flow rate is dynamically adjusted based on the flow weight and traffic load.

The importance of the above class of problems have motivated the search for practical algorithms that approximate the behavior of the idealized GPS. We refer to such algorithms as packet scheduling algorithms. Packet scheduling algorithms are typically used at the MAC layer. The outline presented below classifies packet scheduling algorithms as being suitable for use over wired links, or wireless links.

A. Packet Scheduling Algorithms for Wired Links.

Fair scheduling disciplines have been the subject of intensive study for wired networks. A number of practical packet schedulers have been proposed. Performance guarantees of practical packet scheduling algorithms provide worst case bounds on the achieved performance relative to GPS; the bounds assume that the incoming traffic to each queue is regulated by, for example, a *leaky bucket* regulator. Examples of such practical packet schedulers include the following:

1. Packet-by-packet GPS (PGPS) [18, 61]: PGPS is the first approximation of the (idealized) GPS algorithm that works at the packet level. The PGPS algorithm achieves good accuracy, but has a relatively high computational complexity.

2. The above aspect motivated the search for more efficient algorithms. As a result the following algorithms have been proposed: *self-clocked fair queueing* (SCFQ) [17, 27], *start-time fair queueing* [28], *worst case fair weighted fair queueing* (WF2Q) [7], and *virtual clock* algorithm [83].

B. Packet Scheduling Algorithms for Wireless Links.

Many researchers noted that packet scheduling algorithms developed for wired links are not suitable for use in mobile wireless networks. In such networks, user

mobility and data errors over wireless channels are two key aspects that have to be considered in designing suitable packet scheduling disciplines.

In this section, we give a brief overview of some research work on wireless packet scheduling algorithms for both TDMA-based networks (where transmission power is not taken into consideration), and CDMA-based networks (where transmission power is taken into consideration).

1. Packet scheduling disciplines for TDMA-based Networks: work in this direction includes the following algorithms:

   (a) *Wireless packet service* (WPS) [53]. In [53], the authors propose an idealized wireless fair queuing (IWFQ) algorithm that assumes knowledge of current wireless channel conditions as well as the ability to instantly tag the incoming packets. In addition, the authors present the WPS algorithm as a practical implementation of the IWFQ. It is shown that WPS closely approximates IWFQ, and it provides a short term bound on throughput and delay at the individual packet level.

   (b) *Wireless fair service* (WFS) [54]: WFS consists of different components (e.g. error free service, and compensation models) in order to achieve different performance bounds on fairness, delay and throughput.

   (c) *Effort-limited fair* (ELF) scheduling [19]: ELF extends the weight fair queueing (WFQ) scheduler by utilizing weights that are dynamically adjusted according to channel conditions and users requirements (e.g., throughput, and residual link capacity).

   (d) Other scheduling algorithms in this class include the work of: [52] (the WCFQ algorithm), [73], and [60] (scheduling with location dependent errors), and [68].

2. Packet scheduling disciplines for CDMA-based Networks: work in this direction includes the following algorithms:

   (a) *Fair packet loss sharing* (FPLS) [35] algorithm: the FPLS algorithm considers the following three QoS parameters: transmission bit error

rate (BER), delay, and packet loss requirements. The design aims at distributing the packet loss fairly among the active users in order to maximize the system utilization. The scheduler selects packets for transmission based on the delay requirements, packet loss requirements, and traffic load characteristics. The scheduler first selects the most urgent packets (MUPs) to be transmitted. In case of insufficient resources, the scheduler has to decide on which packets to be dropped for each user based on the packet loss probability for all users.

(b) Other scheduling algorithms includes the work of: [3] (WISPER algorithm), [80] (CDGPS), and [48].

Reference [10] provides a survey of many of the scheduling techniques mentioned above. In summary, we note that packet schedulers that work at the link layer typically keep accurate information on the amount of service provided to each connection stream. In addition, such schedulers are aware of the wireless channel conditions of the mobile terminal associated with each stream. Hence, their design aims at managing the available bandwidth in a fair, and efficient way. CAC schemes, on the other hand, utilize the available connection level information to regulate the work presented to the packet schedulers by admitting, delaying, and/or dropping the incoming requests.

2.4 QoS Provisioning using an Adaptive Multimedia Framework

In [46] the authors propose a call admission control and bandwidth adaptation algorithms for cellular networks where a fixed number of channels is assumed to be available for serving multimedia connections. The work uses the following performance metrics: cell overload probability, effective utilization, and degradation period ratio.

2.4.1 System Model

In [46] the authors consider wireless/mobile multimedia networks where the bandwidth of an ongoing call can be dynamically adjusted by the wireless network to accommodate for various user conditions. The bandwidth of any call is assigned a discrete value from a set $B = \{b_1, b_2, \ldots, b_n\}$, where $b_i < b_{i+1}$ for all $i = 1, 2, \ldots, n$.

The traffic model used by [46] is described by call arrival, call holding time and cell residence time. Call arrival is generated by a Poisson process with mean rate λ calls/sec. Call holding time is assumed to follow an exponential distribution with mean $\frac{1}{\mu}$ seconds. And, cell residence time is also assumed to follow an exponential distribution with mean $\frac{1}{h}$ seconds, where h is the handoff rate.

The main objective for the devised bandwidth allocation algorithm (BAA) is to minimize the number of calls with bandwidth that is lower than a given target value, denote b_{tar}. A *degraded* call is a call with a bandwidth lower than the specified value b_{tar}. The devised algorithm tries to allocates at least b_{tar} for each user whenever it is possible.

An *overloaded cell* is a cell with one or more degraded calls. Therefore, cell overload probability (P_{CO}) is the sum of the probabilities of being in states where there are one, or more, degraded calls. In [46], P_{qos} is a QoS parameter that specifies an upper bound on the cell overload probability and it is given as a QoS requirement (i.e., P_{CO} should be less than P_{qos}).

A one dimensional cellular array is used to model the home cell and adjacent cells. This model is often used to describe streets and highways. Table 2.1 summarizes the symbols used in the paper.

2.4.2 Call Admission Control

In overloaded cell the number of active calls is greater than the maximum number of users that can each be served at the target data rate (i.e., $N_{th} = \frac{C}{b_{tar}}$). Upon the arrival of every new call, the model estimates the cell overload probability P_{CO} after a prescribed estimation time T. $P_{CO} = \sum_{i=N_{th}+1}^{\infty} P(i)$. At the end of the

Table 2.1: Summary of Mathematical Symbols Used in [46]

Symbols	*Definition*
b_{tar}	Target bandwidth
P_{CO}	Cell overload probability
P_{qos}	a QoS parameter that specifies an upper bound on P_{CO}
P_h	Handoff probability
P_r	Probability of staying in the same cell
n	Number of active calls in the given cell C_n
r	Number of active calls in the right cell C_r
l	Number of active calls in the left cell C_l
T	Estimation time
C	Total bandwidth capacity
N_{th}	Maximum number of active calls allocated b_{tar}
$P(i)$	Probability of having i calls in the cell.
B_A	The available bandwidth in the cell.
B_T	The amount of saved bandwidth as a result of reducing all calls with bandwidth more than b_{tar} to calls with b_{tar} bandwidth.
B_M	The amount of bandwidth saved as a result of reducing all call with bandwidth more than b_{min} into calls with b_{min} bandwidth.

estimation time, if P_{CO} is estimated to be greater than P_{qos} the CAC rejects the call. The estimation takes into account the user handoff from adjacent cells, and the handoff from the given cell to adjacent cells. As mentioned in Chapter 1,

$$B(i;n,p) \simeq \left(\begin{array}{c} n \\ i \end{array} \right) p^i (1-p)^{n-i} \tag{2.2}$$

$$P_{t_0+T}(k) \simeq N\Big(nP_r + (l+r)\frac{P_h}{2}, \sqrt{nP_r(1-P_r) + (l+r)\frac{P_h}{2}(1-\frac{P_h}{2})}\Big) \tag{2.3}$$

The overload probability P_{CO} is approximated by the tail of Gaussian distribution as

$$P_{CO} \simeq \sum_{N_{th}+1}^{l+n+r} P_{t_0+T}(k) \simeq Q\Big(\frac{N_{th} - (nP_r + (l+r)\frac{P_h}{2})}{\sqrt{nP_r(1-P_r) + (l+r)\frac{P_h}{2}(1-\frac{P_h}{2})}}\Big) \tag{2.4}$$

where $Q(.)$ is the integral over the tail of a Gaussian distribution. $Q(.)$ is computed using standard Normal distribution with $z = \frac{N_{th}-\mu}{\sigma}$, where μ is the mean and equals to $nP_r + (l+r)\frac{P_h}{2}$ and, σ is the standard deviation and equals

$$\sqrt{nP_r(1-P_r) + (l+r)\frac{P_h}{2}(1-\frac{P_h}{2})}$$

2.4.3 The Bandwidth Allocation Algorithm (BAA)

The main goal of the BAA is to minimize the number of calls where each call is allocated a bandwidth b_{tar} or less. The BAA algorithm covers the following cases:

1. Bandwidth reduction: occurs when a new call arrives that cannot be served at the target bandwidth b_{tar}. The new call may be originated in the cell under consideration, or a handoff call.

2. Bandwidth expansion: occurs when either an ongoing call terminates or leaves the cell.

The main cases for bandwidth reduction in [46] are:

1. if $(B_A \geq b_{tar})$

2. if $(B_A \leq b_{tar}$, and $B_A + B_T \geq b_{tar})$

3. if $(B_A \geq b_{min}$, and $B_A + B_T \leq b_{tar})$

4. if $(B_A \leq b_{min}$, and $B_A + B_T \geq b_{min})$

5. if $(B_A \leq b_{min}$, and $B_A + B_M \geq b_{min})$

6. else drop/block the call

where, B_A is the available bandwidth in the cell, B_T is amount of saved bandwidth as a result of reducing all calls with bandwidth more than b_{tar} to calls with b_{tar} bandwidth, and B_M is amount of bandwidth saved as a result of reducing all call with bandwidth more than b_{min} into calls with b_{min} bandwidth.

2.4.4 Summary

The work of [46] has introduced a call admission control algorithm for adaptive multimedia in networks where a fixed number of channels is assumed to be available for serving the connections. A bandwidth allocation algorithm has also been

proposed to support the proposed CAC. The new CAC uses the cell overload probability to predict the system state at the end of a prescribed estimation time T. By the end of the T the predicted cell overload probability has been used to accept or reject the new call.

The cell overload probability is estimated using the prescribed value b_{tar} of the target bandwidth. The cell is considered overloaded if one or more calls are assigned less than the target bandwidth. Numerical results show that the cell overload probability upper bound is achieved. Also, simulation results show that the proposed CAC reduces the forced termination probability to a negligible level, and increases the effective utilization as the offered load increases.

2.5 Downlink Scheduling in CDMA Data Network

In [39], the authors consider the problem of scheduling traffic on downlink CDMA networks, assuming knowledge of request sizes. The authors use a performance metric, called connection stretch (or normalized delay) to evaluate performance. Table 2.2 summarizes the symbols used in the paper.

Table 2.2: *Summary of Mathematical Symbols Used in [39]*

Symbols	*Definition*
SINR	Signal to interference plus noise ratio
ϕ	Fraction of BS power required to transmit at rate R
R	Transmission rate
E_b	The received energy per data bit
P_I	Total received interference power
P_{BS}	BS maximum available power
FER	Frame error rate
R_i^{max}	Maximum rate
P_i^{max}	Maximum power assigned to connection i
R_i	Achieved throughput
J_i	Connection i
$\|J_i\|$	Connection size in bits
f_i	Delay for user i ($f_i = c_i - a_i$)
t_i	Connection time
a_i	Arrival time for connection i: first bit arrival time instant
c_i	Completion time for connection i: last bit transmission time instant

Power allocation uses a simplified version of equation 1.3 in Chapter 1. The

power equation suggests two variations of resource management: power control, and rate control. Power control is achieved by fixing the data rate R and varying the required transmission power (the ϕ parameter in Table 2.2) to achieve the data rate R. On the other hand, rate control is achieved by fixing the required transmission power, denoted ϕ, and varying the data rate R. The tradeoff between power and rate control is studied in the paper.

2.5.1 Performance Measures and Results

In [39], connection response time (or delay) is defined as the connection completion time minus the connection arrival time ($f_i = c_i - a_i$). The *stretch* factor is defined as the delay experienced by a connection using a given scheduling algorithm relative to the time duration required to serve a connection to completion in unloaded network. The unloaded system delay is computed by: $t_i = |J_i|/R_i^{max}$.

In [39] the authors consider scheduling algorithms to optimize different objective functions. The functions presented in points 1 and 2 below are related to the quantity $f_i = c_i - a_i$. The functions presented in points 3 and 4 below are related to the stretch of connection i defined as follow:

$$s_i = \frac{f_i}{t_i} = \left(\frac{f_i}{|j_i|}\right)\left(\frac{|j_i|}{t_i}\right) = \frac{R_i^{max}}{R_i} \tag{2.5}$$

1. Minimize the maximum delay ($f_i = c_i - a_i$) incurred by connection requests: this function is minimized by first in first out (FIFO) scheduling algorithm.

2. Minimize the average delay ($f_i = c_i - a_i$) over all connection requests: this function is minimized by the shortest remaining processing time (SRPT) scheduling discipline.

3. Minimizing the maximum stretch ($s_i = \frac{f_i}{t_i}$) incurred by connection requests: assuming continuum rates, a near optimal polynomial algorithm exists to minimize this function. Minimizing this function with the assumption of discrete rates is, on the other hand, NP-hard.

4. Minimizing the average stretch ($s_i = \frac{f_i}{t_i}$) over all connection requests: there is no polynomial time algorithm to minimize this function.

A heuristic algorithm, called RMAX, to minimize maximum stretch cost function based on the earliest deadline first (EDF) strategy is proposed to handle both cases of continuum and discrete rates.

A. Algorithm RMAX [39]

1. Assign deadlines $D_i = S.t_i + a_i$ for each queued connection. Where S is a real number such that the stretch of any connection is less than S, t_i is the unloaded system delay.

2. Continuum case: apply the earliest deadline first (EDF) strategy.

3. Discrete case: sort deadlines, then allocate power in order of earliest deadline first, using a greedy algorithm.

4. Upon a connection completion time: update the max-stretch-so-far to the current maximum and the completed connection stretch.

The performance of the proposed algorithm is evaluated against several alternative algorithms, such as FIFO, SRPT, processor sharing and several other algorithms. Different performance measures are used in the evaluation such as average aggregate throughput, average connection stretch, maximum of connection stretch, connection waiting time, and average number of connections which are served at a data rate less than $x\%$ of the maximum rate R^{max}. The results were obtained for both cases: continuum and discrete rates. The experimental results reported in [39] indicate that:

1. Connection size information is useful to improve individual connection satisfaction.

2. The use of both request size, and transmission data rates seems to achieve a good balance between network throughput and individual connection performance.

3. In case of continuous bandwidth, time multiplexing (i.e., use of preemption) outperforms code multiplexing (i.e., change of data rate).

4. In case of discrete bandwidth, a combination of both code and time multiplexing results in satisfactory performance.

5. Continuum rates outperform discrete rates in terms of individual connection satisfaction and response time.

2.5.2 Summary

A class of scheduling problems for downlink data transmission in CDMA networks is studied in [39]. Knowledge of request size and adaptive use of transmission data rates is shown to achieve a balance between networks throughput and individual user satisfaction.

2.6 Dynamic Bandwidth Allocation with Fair Scheduling for WCDMA Networks

In [80] the authors present an algorithm called *code division generalized processor sharing* (CDGPS) scheduling algorithm for handling uplink traffic in CDMA systems. In the system, mobile users are connected to a base station (BS) which is connected to a mobile switching center (MSC). Each user's traffic is sent over a dedicated channel and is assumed to conform to leaky bucket regulator with parameters σ and ρ. Table 2.3 summarizes the symbols used in this paper.

2.6.1 CDGPS Fair Scheduling Scheme

In [80], uplink channel usage time is divided into time slots. The devised algorithm is based on a rate scheduling (i.e., each flow is assigned certain data rate in each time slot) algorithm. At the beginning of each time slot, each active mobile user sends its buffer status to the base station, then based on the following information, the base station determines each user's rate for the next time slot: (1) user QoS requirement (the weight ϕ_i associated with the user's traffic), and (2) available uplink capacity.

Table 2.3: Summary of Mathematical Symbols Used in [80]

Symbols	*Definition*
SIR	Signal to interference ratio
ϕ_i	Pre-assigned weight for user i
$S_i(\tau, t)$	Amount of flow i traffic served during the interval $(\tau, t]$
$C_i(k)$	Flow i capacity during the k^{th} time slot
$B_i(k)$	Total amount of backlogged traffic for flow i during slot k.
$Q_i(\rho_k)$	Backlogged traffic at time ρ_k
$r_i(k)$	Estimated traffic arrival rate of flow i during time slot k
$a_i(k)$	Total amount of traffic arrived in time slot k
t_i^n	Arrival time of the n^{th} packet of flow i
l_i^n	Length of the n^{th} packet of flow i
T	Scheduling period

A. Rate Scheduler

The proposed scheduling algorithm determines the flow rate, denoted $C_i(k)$, for user i at time slot k. The total rate for all flows should not exceed the system capacity, denoted C, at any time slot; we recall that, in CDMA networks, the value C varies depending on the cell state. The devised algorithm is described next. Refer to Table 2.3 for variables description.

Input: Pre-assigned weights for all flows (ϕs).

Output: The assigned service data rate, denoted C_i, to each flow i during the next time slot, such that the vector of all assigned service rates is feasible.

1. Estimate the amount of backlogged traffic of user i at slot k using ($B_i(k) = Q_i(\rho_k) + r_i(k)T$), $i = 1, 2, \ldots, N$. Traffic rate can be estimated by one of the following relations:

   (a) One-step estimation: $r_i(k) = a_i(k-1)/T$.

   (b) Exponential averaging: r_i is updated at the arrival of every new packet. $r_i^{new} = (1 - e^{-T_I^n/K})\frac{l_i^n}{T_i^n} + e^{-T_I^n/K} r_i^{old}$

2. Calculate $S_i(k)$ as follows:

   (a) If $B_i(k) = 0$ then $S_i(k) = 0$

(b) If $B_i(k) > 0$ then $S_i(k) = g_i T$, where

$g_i = \frac{\phi_i C}{\sum_{j=1}^{N} \phi_j}$ is the guaranteed minimum rate for user i, and assigning the data rate $C_i(k) = S_i(K)/T$ to user i is feasible.

In case of $\sum_i S_i(k) < CT$ the remaining network resources are distributed to the users who require more than the minimum guaranteed rate in proportion to each user's weight.

3. $C_i(k) = S_i(K)/T$.

The algorithm assumes that each traffic flow is regulated by a leaky bucket regulator with certain token buffer size, and token generating rate parameters. The paper derives a delay bound in terms of the leaky bucket parameters, and the maximum flow backlog.

2.7 Results

Simulation results show that CDGPS improves the delay performance for heterogeneous traffic (voice, data and video). It also achieves better utilization for the uplink capacity over simpler approaches.

2.8 Summary

GPS scheduling schemes have attracted considerable attention for both wired and wireless networks. The work of [80] presents a new GPS based on dynamic bandwidth allocation scheduling algorithm for the WCDMA networks.

Chapter 3

Admission Control of Delay Bounded Traffic Using Non-preemptive Scheduling

The main results in this chapter are original contributions of the thesis. Specifically, the chapter starts by formalizing a call admission control problem for serving delay bounded traffic in networks where a fixed number of channels is allocated to the traffic. Next, we present a CAC framework for handling the above problem. The CAC utilizes a non-preemptive scheduler as a mechanism for exploiting possible allowable delays associated with individual connection requests to achieve better network utilization. The devised scheduler utilizes dynamic programming.

Performance of the resulting framework is evaluated by simulation in two different contexts. The first context assumes all information on traffic requests are known a priori; this context is suitable for analyzing different traffic traces. The second context does not make the above assumption, and hence it is suitable for real-time systems. A preliminary version of the work in this chapter appears in [44].

3.1 Introduction

As mentioned in Chapter 1, currently, there is an increasing interest in the cellular wireless networking industry in providing mobile subscribers with ubiquitous access to the Internet real time services in a reliable and cost effective way. In this context, provisioning multimedia streaming services to mobile users has received particular attention as an enabling technology for multimedia content playback on mobile terminals with limited energy and storage capacities [56].

Provisioning the required quality-of-service (QoS) measures for streaming services, however, faces the following fundamental design challenges:

1. On the one hand, compared to voice calls, streaming services demand data transmission at relatively higher data rates (e.g., 28 Kbps, or more, before applying channel coding). Provisioning high data rates in cellular networks puts stringent requirements on the use of the available limited wireless resources, and requires the use of sophisticated scheduling mechanisms for the service to be profitable for the network's provider.

2. On the other hand, satisfactory and dependable provisioning of the service to a mobile user require maintaining a session's quality (e.g., data rate, and delays) during the session's lifetime.

Existing approaches to provisioning QoS for real time services in cellular wireless networks rely on the design and use of effective call admission control (CAC) schemes, and scheduling schemes for resource management, as mentioned in Chapters 1 and 2. In this chapter we consider cellular networks where a fixed number of channels are reserved for transmitting the delay sensitive traffic of the streaming QoS class.

One contribution of this chapter is on formalizing a CAC problem for such traffic as an optimization problem that associates two types of delays with each connection request. The associated delays play two key roles in the formalized problem: **(a)** they form the main optimization constraints, and **(b)** they present opportunities of increasing the system utilization by taking suitable scheduling decisions.

The CAC devised in this chapter utilizes a non-preemptive scheduler that exploits the possible opportunities of exploiting individual connection available delays to achieve better system utilization. As noted in Chapter 2, for prior work, we remark that many of the currently well established results on maintaining QoS during inter-cell user movement (handoff) concern voice only traffic (see, e.g., [78]) (c.f. Section 2.1). Research on supporting heterogeneous traffic, and IP-based traffic in cellular networks is more diversified in scope, and objectives, with a vast amount of literature on scheduling physical layer, and link layer frames over wireless channels for both TDMA and CDMA systems. On the other hand, examples of work on scheduling and admission control at higher layers (e.g., connections with known time durations or traffic volumes, transport layer flows and packets, and/or network layer packets) include, for example, the work of [39] (cf. Section 2.5), [46] (cf. Section 2.4), and [81]. The above work, however, does not consider the delay bounds considered in this work.

The rest of the chapter is organized as follows. Sections 3.2 and 3.3 introduce the cellular network model, and give a mathematical formulation of a scheduling problem for serving the underlying streaming requests. Section 3.4 provides some background results and remarks. Sections 3.5, 3.6, and 3.7 describe the architecture of the main components of the proposed adaptive scheduling framework. In section 3.8, we present the obtained performance results of the system, and conclude with some remarks in section 3.9.

3.2 System Model

Throughout this chapter, we consider a GSM-like cellular system with a fixed number of dedicated data channels for serving incoming connection requests of the streaming QoS class. The basic assumptions are:

1. We assume that the wireless network allocates a fixed number of channels, denoted C, for serving the incoming connection requests that require delay bounded services.

2. Users send their multimedia connection requests to the base station. User

requests to the base station for receiving multimedia streams are assumed to be served by multimedia servers located either on the Internet, or hosted by the wireless network provider.

3. In both cases (i.e., multimedia servers located either on the Internet, or hosted by the wireless network provider), the time delay to fetch a requested stream, and make it available for downlink wireless transmission is assumed to consume a negligible amount of time. Hence, user requests that are admitted for service have their arrival time to the base station coinciding with the arrival time of the associated stream of packets from the multimedia server to the base station.

4. Each admitted request is served by allocating one channel. The bandwidth offered by each channel is assumed to be sufficient for serving each multimedia connection at a satisfactory data rate relative to its encoding method and parameters, and hence, there is no need for bandwidth adaptation during a connection's lifetime.

In addition to serving a connection stream at a satisfactory data rate from start to finish, the service model of the streaming QoS class considers the following two types of delays:

- **Start of service delay:** for streaming requests that receive service, this delay is defined as the time elapsed between a user request, and the time the base station starts the downlink transmission. Each request is assumed to be associated with a maximum tolerable start of service delay. Such maximum allowable delay may either be set by the network provider, or explicitly specified by the user's application requesting the stream. The length of the delay period is expected to be longer than the duration of the requested media stream, and is intended to reflect the residence time of a mobile user in one location while waiting for the request to be served.

- **Service interruption delay:** this type of delay may arise if the base station becomes overloaded serving multiple connections that require more than the

allocated number of channels. The arrival of handoff connections is one reason that may give rise to an overload condition.

We now make the following remarks:

1. The system model assumes that all connection requests are served using a single data rate. The single data rate assumption is suitable when all user equipments have similar capabilities.

2. To serve each connection request, the network can utilize one of the following basic service disciplines:

   (a) **Non-preemptive service discipline** in which each connection receives a dedicated data channel from start to completion without interruption.

   (b) **Preemptive service discipline** in which service can be interrupted one, or more, times during a connection's life time.

3. In this chapter, we adopt the non-preemptive service discipline for its simplicity, efficiency, and its potential of minimizing delay jitter at the receiving end during playback.

4. We also note the above two types of delays present both opportunities, and challenges to managing network resources. On the one hand, the CAC can exploit the tolerable start of service delays to admit more traffic connections. On the other hand, the CAC should admit new connections only if they can be served to completion, along with other possible ongoing connections.

We henceforth use the term *delay bounded* traffic to refer to connection requests that have constraints on either the start of service delay, and/or service interruption delay.

3.3 Problem Formulation

As explained above, to best serve the delay bounded traffic, the resource allocation algorithms should take into account the delays associated with each connection.

Dealing with such delays, without interrupting the service to each connection, calls for utilizing a suitable **non-preemptive** scheduling mechanism to decide when each connection should start receiving service.

Such non-preemptive scheduling computations, however, is not suitable for being implemented by the network link-layer packet-scheduler. The architecture considered in this chapter relies on the following ideas:

- dealing with the application level delays of the streaming connections are handled by a call admission control (CAC) module,
- the CAC module utilizes a suitable non-preemptive scheduling module (called the CAC's scheduler) to decide whether to accept, delay, or reject incoming connection requests, and
- the CAC views time as a sequence of fixed-length slots, where each slot is a multiple integer of the link-layer packet scheduler slot length.

We approach the problem of designing an effective CAC scheme to satisfy the design requirements of the streaming class by formalizing the following combinatorial problem. At any instant of time t, the CAC is given the following input:

$R(t) = \{r_1, r_2, r_3, \ldots, r_{n_R}\}$: a set of n_R connection requests. Each request r_i is specified by the following:

- ℓ_i: connection length,
- a_i: arrival instant,
- d_i: maximum allowable delay, and
- w_i: weight.

All intervals and time instants assume integer values in units of the CAC's time slots mentioned above. Request r_i is dropped from the system if it cannot be started on, or before, time $a_i + d_i$.

T_{plan} (the planning interval of the scheduler): T_{plan} is a CAC design parameter, when invoked, the computed schedule allocates channels to requests that start no later than the interval defined by this parameter.

We, now comment on how to incorporate the streaming QoS class delays mentioned above into the above problem framework:

1. For a connection r_i originating in the target cell, we set d_i to the maximum tolerable start of service delay associated with the request.

2. In contrast, if r_i is a handoff request from a neighboring cell to the target cell, then r_i is an ongoing connection that should experience a limited interruption delay. Hence, we set d_i to the maximum tolerable interruption delay associated with that particular connection.

3. Request weights in the problem formulation are intended to give more elaborate control on the behavior of the scheduler. For example, the weights may be set by the network provider to provide service differentiation among classes of requests. Alternatively, they can be set by a resource management module prior to calling the scheduler to give priority to some requests over others.

We can now define an ideal (short term) CAC as one that given a set R(t) of already arrived requests, the CAC admits a subset of requests $R' \subseteq R(t)$ that maximizes the objective function $\sum_{r_i \in R'} w_i$ such that:

1. all admitted requests can be scheduled to start no later than T_{plan} units of time, and

2. if request r_i is selected for service then r_i starts transmission during the interval $[a_i, a_i + d_i]$, and continues to receive uninterrupted service for the entire duration of ℓ_i while being resident in the target cell.

That is, we seek a *non-preemptive* schedule that maximizes the objective function. Connection requests in R' are then forwarded to the link-layer packet scheduler, together with the computed start of service times. Under ideal user channel conditions, all streams in R' can then be transmitted to their respective users without violating the delay constraints.

3.4 Background Results and Remarks

The above CAC design problem calls for selecting a set of connections of maximum total weight that can be non-preemptively scheduled, along with the currently ongoing connections, so as to satisfy the required delay constraints. This type of non-preemptive scheduling problems is related to the class of *interval* scheduling problems for which various related results appear in the literature, as summarized below

1. The above problem has been shown to be NP-complete when T_{plan} is unconstrained, and connection durations, and allowable delays can assume arbitrary values, even when the number of channels $C = 1$. In this special case, the problem is equivalent to problem [SS1] (sequencing with release times, and deadlines) in [26]. The problem is also NP-complete when $C > 1$, and all deadlines are equal (cf. problem [SS8] on multiprocessor scheduling in [26]).

2. Approximation algorithms with bounded approximation ratios for variants of the above problem are given in [5]. In [5] the authors consider a non-preemptive interval scheduling problem where each task is described by release time, deadline, weight, and processing time on each processor. Similar to our problem mentioned above, the objective of [5] is to maximize the total of the served tasks. [5] presents a 5.828 approximation algorithm for solving the problem.

3. We also remark that, for the special case when all tolerable delays are zero, the problem admits efficient solutions for any fixed number of channels C (see, for example, [11, 23] and the references therein).

4. When $T_{plan} < 1$ (i.e., all selected requests must be eligible for receiving service immediately), the optimum solution can be simply obtained by sorting requests that are eligible for receiving immediate service in a non-increasing order of the weights, and greedily serving as many requests as possible according to the sorted list. This approach is referred to as *priority* based

scheduling (PBS), and has been used in many systems (see, e.g., [41] for an overview of the use of some priority rules in scheduling).

5. When $T_{plan} > 1$, a PBS can be used to schedule an arbitrary number of requests, with arbitrary arrival times. The usage scheme involves invoking the scheduler every time a channel becomes available. The performance obtained by such usage, however, is suboptimal. For example, Figure 3.1 shows three requests all arriving at time $t = 0$. Assuming $C = 2$, and $T_{plan} \geq 4$, one can check that all requests can be scheduled. In contrast, for each of the following possible definitions of request weights, at most two requests can be served by a PBS.

   (a) Setting $w_i = \ell_i$ so as to give longer requests a higher priority.

   (b) Setting $w_i = 1/d_i$ so as to give priority to requests that are close to expiry (as in the earlier deadline first scheme).

   (c) Setting $w_i = \ell_i/d_i$ so as to favor longer requests in case two or more requests have the same expiry time.

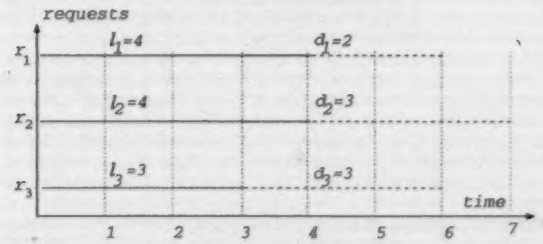

Figure 3.1: *A Scenario with 3 Connection Requests (dashed lines indicate allowable delays).*

3.5 Call Admission Control Framework

In this section, we develop a CAC framework that considers the design requirements mentioned in Section 3.3 for providing delay bounded services for connection requests of the streaming QoS class.

The CAC framework presented in this chapter (refereed to as the delay bounded adaptive system, abbreviated DBAS) has two main modules, refereed to as the T_{plan} *adapter*, and the *CAC's scheduler*. Briefly, the modules perform the following tasks:

1. The T_{plan} adapter acts as a *front* end in the framework; the adapter monitors the incoming traffic requests, and admits requests if free channels are available. When the system is backlogged, the front end also estimates an appropriate planning horizon time T_{plan}, and calls the scheduler (part of the *back* end).

2. The scheduler acts as a *back* end in the framework; the scheduler takes as input the computed T_{plan} time, and executes a heuristic algorithm that schedules a collection of requests, so that each request starts no later than T_{plan} units of time.

Details of each module are presented next.

3.6 T_{plan} Adapter (DBAS Front End)

In the devised framework, when the CAC's scheduler is invoked with a set $R(t)$ of connection requests, the scheduler selects a subset R' of requests for admission, so that each admitted request starts no later that T_{plan} units of time. We now mention the following remarks on the importance of setting the T_{plan} interval:

1. The length of the T_{plan} parameter affects both the amount of work done by the CAC's scheduler, and the set of admitted requests.

2. The use of a simple priority based scheduler (PBS) that does not attempt to set a long term scheduling plan when invoked is advantageous when there is plenty of incoming requests to the system such that the dropped requests are replaced quickly with new arrivals.

3. On the other hand, the use of a scheduler that strives to maximize the objective function by computing a detailed time schedule over a given planning

interval T_{plan} is advantageous when the incoming traffic requests is bursty, with average burst inter-arrival time close to T_{plan}.

4. Moreover, the optimal decisions at various stages depend on the time varying arrival patterns of the streaming requests.

We conclude that the T_{plan} interval affects both the amount of work done by the CAC's scheduler, and the quality of the computed output. Figure 3.2 presents an algorithm for adjusting the T_{plan} interval. Roughly speaking, the function monitors the incoming connection requests, and increases T_{plan} as it observes a decline in the number of arriving requests. In contrast, the function decreases T_{plan} as it observes an increase in the number of arriving requests. To describe the function, we first introduce the following notations:

Q: a queue storing the traffic requests awaiting service.

QD: a queue storing the amount of tolerable delays associated with the most recent C requests for the purpose of computing the average delay $\overline{d}$ associated with the most recent arrivals.

$\overline{d}$: the average tolerable delay of the most recent C arrivals (computed from the data stored in queue QD).

t: the current system time.

t_{SLC}: (time scheduler last called) t_{SLC} is set to zero on initialization, and when the queue Q empties. Otherwise, t_{SLC} is updated every time the scheduler is invoked.

Δ_{SLC}: (time since the scheduler last called) this quantity is defined if $t_{SLC} > 0$ as $\Delta_{SLC} = t - t_{SLC}$.

Figure 3.2 presents a pseudo code of the T_{plan} Adapter module. The T_{plan} Adapter module consists of two phases. The operations of the module can be summarized as follows:

1. The initialization phase sets $t_{SLC} = -1$ to indicate that the scheduler has not been called yet, and clears the queue QD.

2. The main phase is an event driven loop, with request arrivals and completions being the main events. Upon the arrival of new request, the request's tolerable delay is added to QD; If the request cannot be served (all channels are busy) then add the request to queue Q, else serve the request; The main events of the loop are:

   - When a request completes service and the scheduler has not been called since the last time Q has emptied (i.e., $(t_{SLC} = -1)$, if $(|Q| = 1)$ serve the queued request; else if $(|Q| \geq 2)$ compute the average tolerable delay $\overline{d}$ from the data in QD; T_{plan} is set to $\overline{d}$, and finally the scheduler is called. Here, $\overline{d}$ has been selected as a convenient initial value.
   - On request completion event and $t_{SLC} > 0$, the module considers different actions according to Q's size. If $(|Q| = 0)$ then reset $t_{SLC} = -1$; if $(|Q| = 1)$ serve the queued request; Finally, for the the case of $(|Q| \geq 2)$, the module considers three cases. First case: no arrival since last scheduler invocation, and T_{plan} has not expired (i.e., $T_{plan} > \Delta_{SLC}$), this indicates that the current value of T_{plan} is adequate, and no further action is required. Second case: one, or more, arrivals occurred since last scheduler invocation, and T_{plan} has not expired, the scheduler is invoked, and T_{plan} is shortened using exponential moving average: $T_{plan} = \alpha T_{plan} + (1 - \alpha)\Delta_{SLC}$, we use $\alpha = 0.5$ in the simulation study. Third case: T_{plan} expired (i.e., $T_{plan} \leq \Delta_{SLC}$), the current value of T_{plan} is perceived to be inadequately short; extend T_{plan} gradually, up to a maximum value of the longest tolerable delay of requests awaiting service.

We conclude this section by remarking that when $T_{plan} < 1$, both a PBS and the main scheduling function of the CAC (cf. Figure 3.6) compute an optimum solution. The adaptive scheme is designed to converge to using short T_{plan} intervals when it detects a traffic pattern where dropped requests are quickly replaced with

T_{plan} Adapter (DBAS Front End):

1. Initialize $t_{SLC} = -1$, and $QD = \phi$
2. do forever {

switch (event)

event: arrival of a new request.

- add the tolerable delay of the request to QD;
- if the request cannot be served (all channels are busy) then add the request to queue Q, else serve the request;
- break;

event: completion of a request, and $(t_{SLC} == -1)$

- if $(|Q| == 1)$ serve the queued request;

else if $(|Q| \geq 2)$ compute the average tolerable delay $\overline{d}$ from the data in QD; set $T_{plan} = \overline{d}$; call the scheduler;

- break;

event: completion of a request, and $(t_{SLC} > 0)$

- if $(|Q| == 0)$ reset $t_{SLC} = -1$;

else if $(|Q| == 1)$ serve the queued request;

else if $(|Q| \geq 2)$ {

case: no arrival since last scheduler invocation, and T_{plan} has not expired (i.e., $T_{plan} > \Delta_{SLC}$).

The current value of T_{plan} is viewed as adequate; continue executing the remaining part of the scheduled requests;

break;

case: one, or more, arrivals occurred since last scheduler invocation, and T_{plan} has not expired.

The arrival of new requests require invoking the scheduler; the current value of T_{plan} is perceived to be longer than needed. To shorten the value, use an exponential moving average: $T_{plan} = \alpha T_{plan} + (1 - \alpha)\Delta_{SLC}$;

break;

case: T_{plan} expired (i.e., $T_{plan} \leq \Delta_{SLC}$).

The current value of T_{plan} is perceived to be inadequately short; extend T_{plan} gradually, up to a maximum value of the longest tolerable delay of requests awaiting service;

break;

}

break;

}

Figure 3.2: *Pseudo-code of the T_{plan} Adapter of the Delay Bounded Adaptive Scheduling (DBAS) Framework.*

new requests. Hence, the devised DBAS framework is expected to be competitive with a PBS even for traffic patterns that suit PBS best.

3.7 The Scheduler (DBAS Back End)

The scheduling module is a core module in the proposed framework. When invoked at time t, the scheduler takes as input the following parameters:

1. the number of free channels at time t, denoted C,
2. a set $R(t) = \{r_1, r_2, ..., r_{nR}\}$ of outstanding unexpired connection requests, each request r_i is described by its duration ℓ_i, and the (possibly updated) values of its weight w_i, and delay constraint, denoted d_i, and
3. the current value of the T_{plan} parameter.

An ideal scheduler selects a subset $R' \in R(t)$ of maximum total weight $\sum_{r_i \in R'} w_i$ that can be scheduled **non-preemptively** on the $C(t)$ channels, so that no connection starts later than $t + T_{plan}$.

As remarked in section 3.4, if T_{plan}, connection lengths, and tolerable delays are allowed to assume arbitrary values, the problem of designing an ideal scheduler is NP-complete (even for $C = 1$). In this section, we devise a heuristic scheduling model based on the idea of *k-channel packing* (k-CPS) described below.

3.7.1 Design of k-Channel Packing Schedulers

To start, we call a scheduler that computes a schedule for a given number k of channels a *k-channel packing* scheduler (k-CPS). The availability of a k-CPS algorithm (for any $k \geq 1$) allows us to design a heuristic algorithm to construct a schedule for any number C, $C \geq k$, of channels. The idea is to invoke such algorithm $\lceil C/k \rceil$ times. Each invocation, selects a subset of connection requests that can be scheduled over k channels (the last invocation selects connection requests that can be scheduled over $\leq k$ channels), the selected connections are subsequently removed from the input. The following remarks apply to the above approach:

1. For $k = 1$, the construction of an optimal 1-CPS is conceptually simple.

2. The case of developing and utilizing a 2-CPS is more interesting, nevertheless, we are not aware of any performance study of the resulting class of schedulers. One contribution in this chapter is to develop a dynamic program for an optimal 2-CPS, and examine its performance.

3. We note that for any value of $k \geq 1$, the performance of a $(k+1)$-CPS can be shown to be better, than the performance obtained by applying an optimal k-CPS, followed by an optimal 1-CPS. To illustrate this point, consider, the following example.

   Example.

   The problem instance in Figure 3.3 has five requests, where $C = 2$ channels, $T_{plan} \geq 8$, the weight of each request is its length (e.g., $w_1 = \ell_1 = 4$), and the tolerable delay is given by the length of the associated dashed line (e.g., $d_1 = 1$). Two consecutive applications of an optimal 1-CPS may produce the sequences (r_3, r_2, r_5), and (r_1) (or, alternatively, (r_4) instead of (r_1)) that form a maximal schedule over $C = 2$ channels of total weight = 14 units. In contrast, an optimal 2-CPS computes the sequences (r_1, r_2), and (r_3, r_4, r_5) of total weight = 18 units. In this example, a 2-CPS achieves more than 28% gain than repeated application of a 1-CPS. □

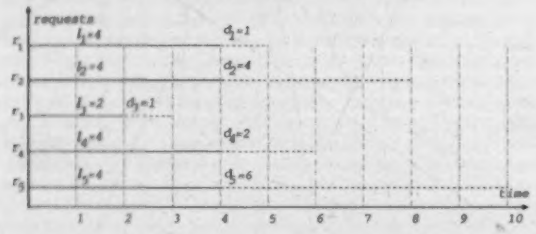

***Figure 3.3:** A Scenario with Five Connection Requests.*

3.7.2 Scheduling Algorithm

In the above section, it has been shown that an optimal 2-CPS can achieve a substantial improvement over the iterated use of an optimal 1-CPS. The availability of such optimal 2-CPS can then be used to construct a scheduler for any number of channels C, $C > 2$, using the setup outlined in Figure 3.4. Below, we devise one such novel 2-CPS, and analyze its running time.

The Scheduler (DBAS Back End):
Input: T_{plan}, and the list of queued requests in Q
Output: a subset R' of requests of $R(t)$ that can all be started within T_{plan} units of time

1. let S = the subset of requests whose remaining tolerable delay $\leq T_{plan}$.
2. for $i = 0, 1, \ldots, \lfloor \frac{C}{2} \rfloor - 1$
   invoke function 2-CPS to schedule requests to be served by channels $2i$, and $2i+1$;
   remove the scheduled connections from S;
3. if $(2(i+1) < C)$ obtain an optimum packing for channel C;
4. start executing the scheduled requests

Figure 3.4: *Pseudo-code of the Scheduler of the DBAS Framework.*

3.7.3 Two-Channel Packing Scheduler

Function 2-CPS described below is the core module that computes an optimum solution for packing traffic requests into two channels. The design outlined below is intended to be general and useful for both online usage, and offline usage to analyze scheduling of traffic traces:

- Online scheduling usage is our present context where all traffic requests are assumed to have arrived to the base station before invoking the scheduler.
- Off-line scheduling usage is useful when running the back end module as a stand alone tool for the purpose of analyzing traces of traffic requests where arrivals can appear anywhere inside the trace.

The above generalization in the overall design is achieved with a negligible increase in the implementation complexity.

The following summarizes the important additional variables used by the function. To simplify the presentation, throughout this section we treat the time instant when the scheduler is invoked as $t = 0$. All traffic requests arrival times, and remaining tolerable delay intervals are relative to this time instant.

$DR = \{r_{i,d} | \, r_i \in R, \text{and } d = 0, 1, \ldots, \min(d_i, T_{plan})\}$: the set of all undelayed, and delayed (up to a maximum of T_{plan} units) instances of traffic requests.

n_{DR}: the number of requests in DR. That is, $n_{DR} = \sum_{r_i \in R}(1 + \min(d_i, T_{plan}))$, where the term "1" accounts for the undelayed version of request r_i, and the term $\min(d_i, T_{plan})$ accounts for the number of possible delayed versions of request r_i.

$L[1, 2, \ldots, n_{DR}]$: a list of the traffic instances in DR sorted in a nondecreasing order of their starting times. That is, for request indexes i and j, the instance $r_{i,d}$ comes before the instance $r_{j,d'}$ in L if $d \leq d'$.

$L[k]$: For simplicity of presentation, if k is a valid index in L corresponding to an instance $r_{j,d}$ of request $r_j \in R$, then $L[k]$ refers to the identity (i.e. the index j) of that request. So $w_{L[k]}$ refers to the weight w_j of r_j.

Example.

Figure 3.5 illustrates 5 traffic requests $(r_1, r_2, \ldots, r_5)$ arriving at $t = 0$. The set DR of all undelayed, and delayed request instances has $n_{DR} =$ 19 instances appearing in the list L shown below. A possible list L of traffic instances stored in a non-decreasing order of their starting time is

$$\begin{aligned} L = \; & (r_{1,0}, r_{2,0}, \ldots, r_{5,0}, r_{1,1}, r_{2,1}, \ldots, r_{5,1}, \\ & r_{2,2}, r_{4,2}, r_{5,2}, r_{2,3}, r_{5,3}, \\ & r_{2,4}, r_{5,4}, r_{5,5}, r_{5,6}) \end{aligned}$$

For $k = 7$, the k^{th} element in L is $r_{2,1}$. Using the above mentioned convention, we get $L[7]$ equals the identity of request $r_{2,1}$, which is r_2. So $w_{L[7]}$ refers to w_2. □

A. Variables of the Dynamic Program

The devised scheduler computes an optimum solution using a dynamic program that considers all instances of requests in DR according to the ordering determined by L. In particular, at the kth stage, $k = 0, 1, 2, \ldots, n_{DR}$, the function considers the effect of including, and excluding, the request $L[k]$, by keeping track of quantities of the types $S^{(k)}[i, j]$ described below. In the following definition, i and j are assumed to be time instants where $i \leq j$, and $i, j \in [-1, T_{plan}]$. The special value of -1 is used to denote a boundary condition that does not correspond to any possible instant when a request can be scheduled to start receiving service.

$S^{(k)}[i, j]$: a subset of $\{L[q] | \, q = 1, 2, \ldots, k\}$ (i.e., the first k entries of L) is of type $S^{(k)}[i, j]$ if

1. no two instances in the subset correspond to the same request in R, and
2. the subset can be scheduled over two channels such that the latest requests scheduled for service over the two channels start at instants i, and j, respectively, where $i \leq j$.

Note that the special case where $i = -1$ indicates that no request is scheduled for service on one channel. In addition, the special case where $i = j = -1$ indicates that no request is scheduled for service over both channels.

Example.

in Figure 3.5, if we let

$$L = (r_{1,0}, r_{2,0}, \ldots, r_{5,0}, r_{1,1}, r_{2,1}, \ldots, r_{5,1}, r_{2,2}, r_{4,2}, r_{5,2}, r_{2,3}, r_{5,3}, r_{2,4}, r_{5,4}, r_{5,5}, r_{5,6})$$

then the set $\{r_{1,0}, r_{3,0}, r_{4,2}\}$ is of type $S^{(12)}[i = 0, j = 2]$ (it may also be of other types) since all three instances belong to the first 12 entries of L, and the instances can be scheduled over two channels such that

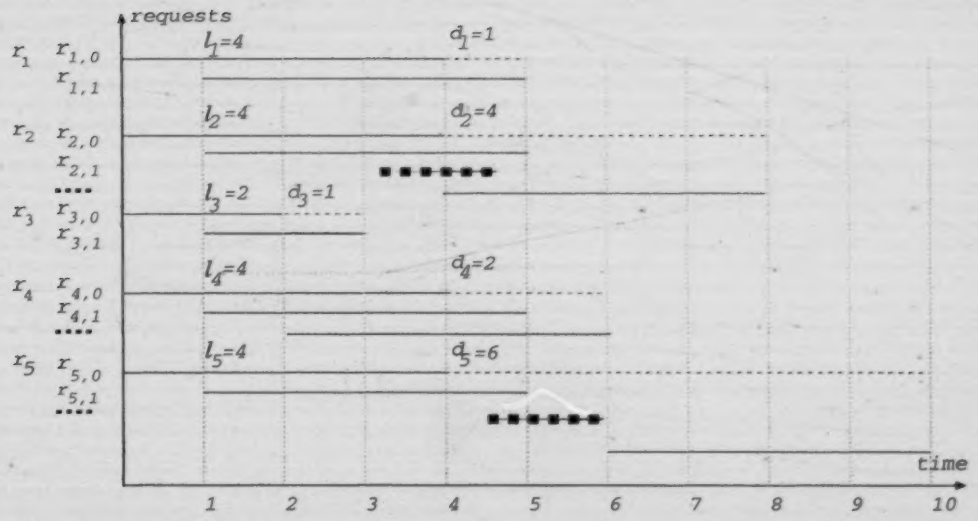

Figure 3.5: *The set DR of all Undelayed, and Delayed Connection Request Instances of the Set R in Figure 3.3 (a total of $2+5+2+3+7=19$ instances).*

$r_{1,0}$ is packed in one channel (starts at time $i = 0$), and $\{r_{3,0}, r_{4,2}\}$ are packed into the second channel (request 4 starts at time $j = 2$).

And, the set $\{r_{1,0}\}$ is of type $S^{(1)}[i = -1, j = 0]$ since $r_{1,0}$ is the first entry in L, and it can be scheduled over two channels such that one channel is empty (so, $i = -1$), and the other channel starts at time $j = 0$. □

In function 2-CPS, we use variables denoted $S^{(k-1)}[i, j]$ to store maximum weighted sets of the corresponding types. Furthermore, we say that request $L[k]$ can be *tightly* packed with set $S^{(k-1)}[i, j]$ if

1. $S^{(k-1)}[i, j]$ can be scheduled over two channels, and
2. if $L[k] = r_{j,d}$ for some request r_j, where $d > 0$ (i.e. $L[k]$ is a positively delayed instance) then there is no idle time between the start of $L[k]$, and the end of it preceding request on the same channel.

B. The Dynamic Program

Figure 3.6 presents the 2-CPS function. The function consists of two parts: the initialization part, and the main loop. The initialization part (steps 1 and 2)

Function 2-CPS:
Input: T_{plan}, and R, as described above
Output: a subset of DR with maximum total weight that can be scheduled over two channels.

1. construct the set DR, and the ordered list L.
2. perform the following initializations:
   case: both channels are not currently serving requests
   set $S^{(0)}[-1,-1] = \phi$; break;
   case: one channel is empty, the other is currently serving a request r
   set $S^{(0)}[-1,0] = \{r\}$; break;
   case: both channels are busy serving two requests r_1 and r_2
   set $S^{(0)}[0,0] = \{r_1, r_2\}$; break;
3. for $k = 1, 2, \ldots, n_{DR}$ {
   3.1. for all relevant pairs of time instants (i,j), $i \leq j$, $i,j \in [0, T_{plan}]$ {
   $S^{(k)}[i,j]$ = maximum weighted subset of $(S^{(k-1)}[i,j]$, $\{S^{(k-1)}[i',j'] \bigcup L[k]$: where $L[k]$ can be tightly packed with $S^{(k-1)}[i',j']$ to obtain a valid $S^{(k)}[i,j]$ set $\})$
   }
4. return a set from the collection $S^{(n_{DR})}[.,.]$ with the maximum possible total weight

***Figure 3.6:** Pseudo-code for Function 2-CPS.*

constructs both sets DR and L, and initializes an array that stores the collection $S^{(0)}[i,j]$ with entries corresponding to the ongoing requests. In step 3, the function processes each connection request instance $L[k]$ for $k = 1, 2, \ldots, n_{DR}$. Processing $L[k]$ is done by constructing array $S^{(k)}[i,j]$ from entries in the previously computed array $S^{(k-1)}[i,j]$

A number of remarks on the implementation, correctness, and timing of the above function now follow.

C. Implementation Issues

We now mention a few remarks on the implementation aspects of the function 2-CPS.

- As can be seen from Figure 3.6, the collection of sets computed at any stage k,

$k > 1$, is computed from the collection of sets pre-computed at the previous stage $k-1$. Thus , the above function can be implemented with just two arrays of sets, where entries in each array are indexed by a pair (i, j) of time instants.

- Furthermore, the relevant entries in the array storing the collection $S^{(k)}[.,.]$ of sets can be computed as the need arises when processing the entries of the array storing the collection $S^{(k-1)}[.,.]$. That is, there is no need to exhaustively consider all possible pairs (i, j) where $i \leq j$, and $i, j \in \{-1, 0, 1, \ldots, T_{plan}\}$ since many combinations will correspond to empty sets. The length of each such table, however, remains bounded by $O(\frac{1}{2}{T_{plan}}^2)$ (since $i \leq j$).

D. Correctness

The above function can be shown to be correct since the program considers both the effect of including, and excluding each eligible instance in the set DR. In particular, consideration of a set $S^{(k-1)}[i, j]$ in step 3.1, accounts for the case where the instance $L[k]$ is excluded from computing a possible maximum weighted $S^{(k)}[i, j]$ set. On the other hand, consideration of sets of type $S^{(k-1)}[i', j'] \bigcup L[k]$ accounts for cases where $L[k]$ are included in computing a possible maximum weighted $S^{(k)}[i, j]$ set.

E. Running Time

The function performs n_{DR} iterations. If ℓ_{min} is the length of the shortest request in R, then in each iteration, the length of each array described above is bounded by $O(\frac{1}{2}(\frac{T_{plan}}{\ell_{min}})^2)$. So, the total running time is $O(\frac{1}{2}(\frac{T_{plan}}{\ell_{min}})^2)n_{DR})$.

3.7.4 One-channel Packing Scheduler

As mentioned above, if the number of channels C is an odd integer, then step 3 of Figure 3.4 seeks to compute an optimum packing of the last (unpaired) channel C. We remark that function 2-CPS can be used to construct an optimal schedule over

one channel (for example, by assuming that one of the two channels has an ongoing connection request of length $\geq T_{plan}$).

The running time in this type of invocation is $O((\frac{T_{plan}}{\ell_{min}}))n_{DR})$, where $\frac{T_{plan}}{L_{min}}$ is the length of each array mentioned in the implementation section. A more careful implementation of the function, however, reveals that the function can be implemented to run in $O(T_{plan}n_R)$, where n_R is the number of undelayed requests in the input set $R(t)$.

3.8 Performance Results

In this section, we use simulation to explore the usefulness and effectiveness of the proposed design concepts. In particular we consider the following aspects:

1. **Evaluating the exponential moving average effect**: as presented in Section 3.6, the T_{plan} adapter module utilizes an exponential moving average mechanism with parameter, denoted α. We explore in Section 3.8.1 the effect of varying the parameter α on the system's effective throughput. To explore this particular aspect, a complete implementation of the entire CAC system is utilized.

2. **Evaluating the CAC's scheduler performance**: to assess the effectiveness of the scheduling module, we use the module in an offline context where a given traffic trace is analyzed. The obtained results are compared to the results achieved by using a predictive CAC obtained by modifying an algorithm due to [46].

3. **Evaluating the overall framework performance**: here, we assess the performance the overall framework in an online context where the comparison is done against a priority based scheduler (PBS).

Throughout this section, user connection requests, and mobility parameters are described by the following parameters (see also [46]).

λ: (connection requests arrival rate) streaming connection requests are assumed to originate in the target cell, and the neighboring cells, as a Poisson process with rate λ connection/sec.

$1/\mu$: (average connection duration) the time duration of each connection request is assumed to be exponentially distributed with an average of $1/\mu$ seconds.

$1/\delta$: (average connection residence time) the time duration during which a connection is served in its originating cell (before moving to a neighboring cell) is assumed to be exponentially distributed with average $1/\delta$ seconds. After that time duration, the ongoing connection is assumed to handoff to one of the neighboring cell according to some probability distribution.

3.8.1 Evaluating the Exponential Moving Average Effect

As presented in Figure 3.2, the T_{plan} adapter utilizes an exponential moving average (EMA) method to shorten the planning period using the formula:

$$T_{plan} = \alpha.T_{plan} + (1 - \alpha).\Delta_{SLC}. \tag{3.1}$$

In the formula, the most recent computed T_{plan} values are favored by using a large value for the parameter α. To explore the effect of the parameter α on the overall performance of the system, several simulation experiments are conducted using a complete implementation of the CAC module.

Figure 3.7 illustrates the results obtained by changing the parameter α in the range $\alpha = 0.0, 0.1, 0.5, 0.9, 1.0$, and the offered traffic load in the range 7, 10, 15, 17, and obtaining the resulting effective system throughput. Numerical values of other parameters related to user mobility are set as in Section 3.8.2. We observe that the resulting effective throughput is not particularly sensitive to changes in α. So, we set $\alpha = 0.5$ for the rest of the experiments.

3.8.2 Evaluating the Scheduler Performance

As mentioned above, the scheduler is a core module in the devised DBAS framework. For the purpose of evaluating the performance of the scheduler we conduct

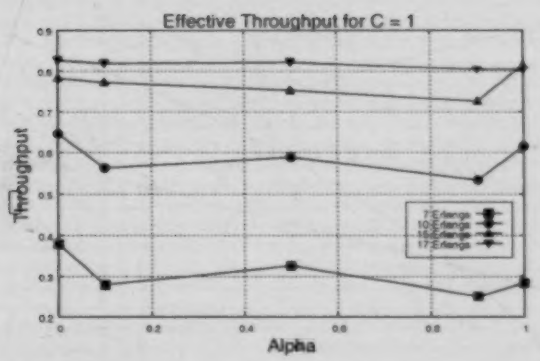

***Figure 3.7:** Effect of Choosing the Parameter α in the EMA on the Effective Throughput Under Different Offered Traffic Load.*

simulation experiments under the following assumptions:

1. The DBAS CAC module is invoked with all traffic request information known a priori, and unconstrained value of T_{plan}.

2. The performance of the DBAS framework is compared against a CAC scheme, referred to below as the *predictive CAC*. The predictive CAC scheme introduces modifications to the CAC scheme in [46].

3. In contrast to the DBAS invocation assumption mentioned above, at any instant t, the devised predictive CAC is invoked with information on traffic requests that have arrived on, or before, the instant t. However, as mentioned below, the predictive CAC assumes a priori knowledge of traffic distributions (e.g., distribution of connection request arrival times, distribution of connection duration times).

4. The predictive CAC is designed to give higher priority to the handoff traffic, and hence, the predictive algorithm is expected to be particularly effective in minimizing the forced termination probability.

Below, we first start be describing the predictive CAC scheme, followed by presenting and discussing the obtained results.

A. The predictive CAC

We recall that in [46], the authors introduced a predictive CAC for serving streaming connections in a system with the following properties:

- The bandwidth of an ongoing multimedia connection can be dynamically adjusted several times during the connection's lifetime. In particular, the assigned data rate to a connection can be decreased if the target cell becomes congested.
- The CAC devised in [46] works by predicting a cell *overload* probability, denoted P_{CO}, at the end of some estimation time T; P_{CO} is defined as the probability that the number of ongoing connections in the target cell at the end of the interval T exceeds the maximum number of connections that can be served at a target data rate.
- The CAC rejects a connection request if the computed P_{CO} exceeds a prescribed threshold probability, denoted P_{qos}.

Predictive CAC:
Input: traffic, and mobility parameters, as described above, as well as CAC control parameters T and P_{qos}.

1. **the CAC queue:**
   - upon arrival of a handoff connection, admit the connection by adding it to the scheduler's queue;
   - upon arrival of a connection originating in the target cell (and every fixed interval of time thereafter) get the number of active connections in the target cell, and its neighbors, use equation (3) in [46] to evaluate P_{CO}; if the computed $P_{CO} \leq P_{qos}$ admit the connection with highest weight by moving it to the scheduler's queue;
2. **the CAC's scheduler's queue:** use a priority based scheduler to manage the scheduler's queue.
3. remove expired connections periodically

***Figure 3.8:** Pseudo-code of a Predictive Admission Control Scheme.*

In this section, we use a modified version of the above predictive CAC mechanism adapted to serve the present context where each admitted connection request

occupies one channel during its lifetime, and connections are assumed to be served non-preemptively. In the modified version, the CAC adds an incoming handoff connection to the scheduling queue without subjecting it to the $P_{CO} \leq P_{qos}$ test.

In contrast, a connection request originating in the target cell is delayed until the computed P_{CO} falls below P_{qos}. Thus, the modified version gives higher priority to the handoff traffic, and hence, the predictive algorithm is expected to be particularly effective in minimizing the forced termination probability.

B. Simulation Parameters

The numerical values used to derive the results are summarized in Table 3.1. System capacity C is set to 200 channels, each user is allocated 10 channels at each time, thus, the system can support a maximum of 20 users at a time. For each connection request r_i, we set $d_i = 0.1\ell_i$. The arrival rate λ is varied to generate different levels of offered traffic loads.

***Table 3.1:** Simulation Parameters*

Parameter	Value
C	200 channels
$1/\mu$	200 seconds
$1/\delta$	100 seconds
T	20 seconds
P_{qos}	0.2
d_i	$0.1\ell_i$

C. Numerical Results

Throughout the thesis, each simulation run is repeated a number of times to achieve a 95% confidence level; the average of such runs is depicted in the figures in chapters 3, 4, and 5. To avoid transient effects, each experiment is run for a sufficiently long time.

1. Effective Throughput

Figure 3.9 depicts the effective throughput results. As the system load increases, the effective throughput increases for both schemes. The proposed DBAS scheme

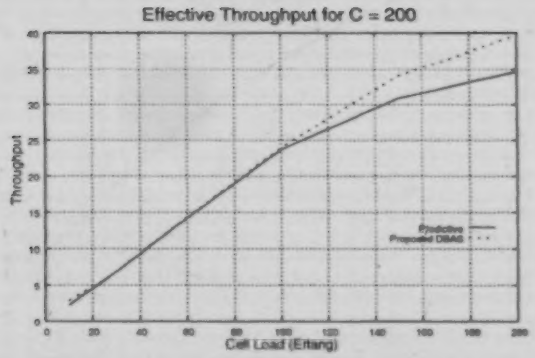

Figure 3.9: *Throughput of the Adaptive DBAS Framework versus the Predictive CAC [C=200].*

achieves higher throughput at high cell loads. The obtained results suggest that the DBAS scheme is successful in achieving its main objective function. At low cell load, both schemes achieve very close results.

2. Percentages of Forcibly Terminated Connections

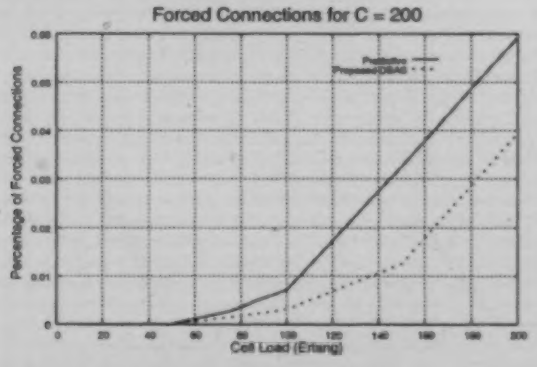

Figure 3.10: *Forced Connections of the Adaptive DBAS Framework versus the Predictive CAC [C=200].*

The percentages of forcibly terminated connections results are shown in Figure 3.10. The proposed DBAS scheme outperforms the predictive scheme in terms of the percentage of forcibly terminated connections. both schemes managed to achieve relatively low percentage of forcibly terminated connections (less than 6% in case of predictive scheme, and less than 4% in case of DBAS scheme).

3. Percentages of Completed Connections

Figure 3.11 shows the results for the percentage of completed connections. It shows that increasing the offered load (in Erlang) decreases the percentage of completed connections. It also shows that the DBAS scheme serves more connections to completion than the predictive scheme.

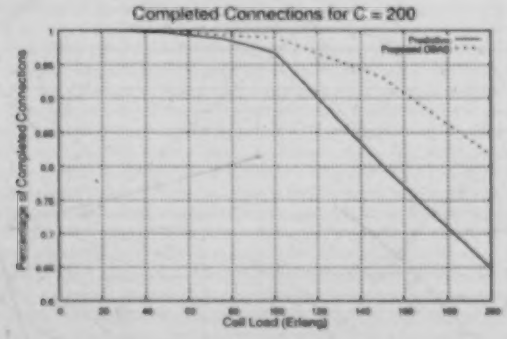

Figure 3.11: *Completed Connections of the Adaptive DBAS Framework versus the Predictive CAC [C=200].*

4. Percentages of Blocked Connections

Figure 3.12 depicts the results for the percentages of blocked connections. Results show that DBAS framework outperforms the adaptive scheme in terms of number of blocked connections. Increasing the cell load increases the percentage of blocked calls.

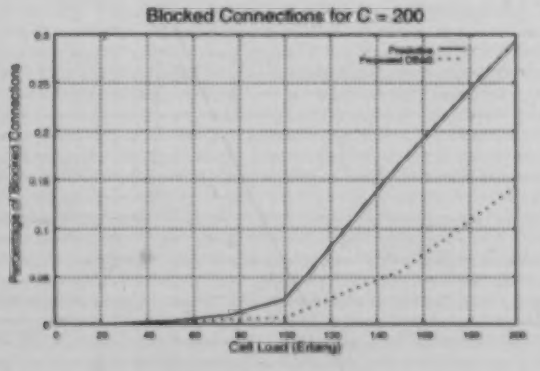

Figure 3.12: *Blocked Connections of the Adaptive DBAS Framework versus the Predictive CAC [C=200].*

5. Remarks.

The above figures show that the proposed DBAS framework outperforms the predictive algorithm. It achieves higher effective throughput, higher percentage of completed connections and lower percentage of blocked and forced terminated connections. This difference in performance is rather obvious at high offered cell loads. Both algorithms achieved relatively similar performance at low offered cell load.

The predictive algorithm accepts new connections (newly originated in the cell or handoff) up to a certain threshold, this threshold depends on the system capacity and the overload probability estimation formula proposed in [46]. As the algorithm accepts handoff connections without subjecting them to the overload test, it is possible that a low priority (i.e., small weight) handoff connections get admission preference over a newly generated connection with higher priority. On the other hand, the DBAS framework deals with handoff traffic by assigning higher weight function to the handoff connections. Applying this technique ensures a fair treatment to handoff connections and at the same time prevents low priority handoff connections from being admitted over new higher priority connections.

The obtained results indicates that the predictive scheme succeeds in maintaining a low percentage of forcibly terminated connections for all tested values of the offered traffic load. The use of the proposed DBAS framework, nevertheless, indicates a room for further improvement. Figures 3.12, and 3.9, taken together, indicate that such improvement can also be achieved while increasing the effective throughput, at the expense of a slight increase in the blocking of new connection requests originating in the target cell.

3.8.3 Evaluating the Overall Framework Performance

The second set of experiments compares the performance of the DBAS framework with a CAC that utilizes a priority based scheduling scheme (PBS) in an online context. In both cases, a connection request r_i has a weight of $w_i = \ell_i/d_i$. Figure 3.13 illustrates the achieved average throughput per channel in both cases. Here, we examine the performance of the DBAS framework utilizing both the scheduler and

the T_{plan} adapter. As can be seen, the devised framework consistently outperforms the PBS for all tested values of the offered load. As the system load increases, the effective throughput for both schemes increases.

As seen in the previous section, the use of DBAS's scheduler alone (that is suitable for trace analysis context) is shown to achieve a good performance. As this framework (the use of both the scheduler and the T_{plan} adapter) is more suitable for normal (online) context, its is also shown that the proposed framework outperforms another commonly used scheme (PBS).

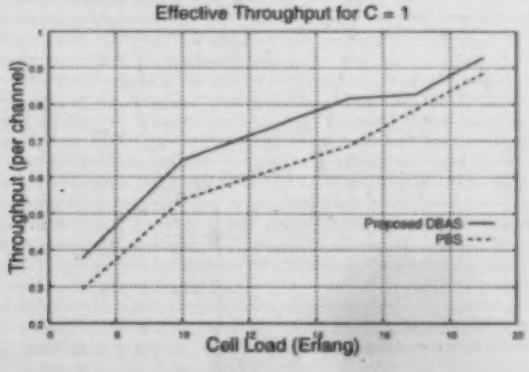

Figure 3.13: *Effective Throughput (per channel) of the DBAS Framework versus a Priority Based Scheduler.*

3.9 Concluding Remarks

In this chapter we investigate the design of a call admission control framework for managing multimedia connection requests in a wireless cellular network where the streaming QoS service class is allocated a fixed number of dedicated data channels. The proposed design utilizes non-preemptive scheduler to exploit per-connection possible allowable delay opportunities to achieve higher throughput. The architecture combines the use of an adaptive approach to deduce a suitable planning interval that guides the working of the CAC's scheduler, with a novel scheduling heuristic based on devising an optimum 2-channel packing dynamic programming scheduler. The reported results indicate the usefulness of the proposed approach when applied in both online and offline contexts.

Chapter 4

Admission Control of Delay Bounded Traffic Using Preemptive Scheduling

In this chapter we consider a framework similar to the framework presented in Chapter 3 for admission control of delay bounded traffic in networks where a fixed number of channels is dedicated for the traffic.

Rather than using a non-preemptive scheduler in the CAC design as done in Chapter 3, a novel contribution of this chapter is on devising a preemptive scheduler. The task of redesigning the CAC using a preemptive scheduler introduces complications in the scheduler's design, as well as complications in dealing with the delay constraints (the start of service delay, and the service interruption delay).

A heuristic algorithm for handling the preemptive scheduling problem is devised, and integrated with the mechanism introduced in Chapter 3 for dynamically adapting the scheduling planning interval according to the offered connection request pattern. The devised framework does not assume a priori knowledge of the distribution of the arriving requests from either the target cell, or as a consequence of handoff.

Performance is compared against an admission control scheme that assumes a priori knowledge of such traffic distributions (e.g., distribution of connection request arrival times, distribution of connection duration times, etc.), and admits a connection only if the estimated cell overload probability, after a prescribed prediction interval, does not ex-

ceed a specified threshold value. The obtained results show improvements with regard to the achieved throughput of the connections served to completion, and forced terminations. A preliminary version of the work presented in this chapter is documented in [45].

4.1 Introduction

The work presented in this chapter is motivated by the good performance results obtained in Chapter 3. One way of enhancing the adaptive framework of Chapter 3 is to replace the non-preemptive scheduler with a preemptive scheduler, where the service of any connection request can be preempted several times during the connection life time.

To illustrate the potential gain from using a preemptive scheduler we consider the following simple example.

Example.

Figure 4.1 illustrates three connection requests with the following durations, and acceptable delay values parameters: $(l_1 = 200, d_1 = 100), (l_2 = 200, d_2 = 100), and (l_3 = 200, d_3 = 100)$ to be served by $C = 2$ channels. A non-preemptive scheme of the requests selects 2 connections, and achieves a throughput of 400 units of service time. A preemptive scheduler selects 3 connections, and achieves throughput of 600 units of service time. □

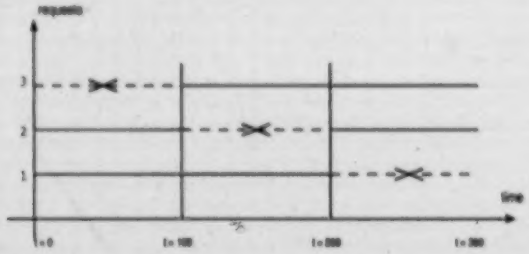

Figure 4.1: *Example: Preemptive versus Non-Preemptive Scheduling*

The introduction of a preemptive scheduler, however, presents some new challenges. In particular, we note the following:

1. In a non-preemptive scheduling context, if a connection request originates in the cell under analysis then the start of service delay, and the connection duration determines the request's deadline time. Similarly, for a handoff connection, the remaining service interruption delay, and the remaining part of the required service time determine the request's deadline time. In both cases, the design of the non-preemptive scheduler is simplified by considering a single deadline time for each request.

   In contrast, for a preemptive scheduler, the two delay types play two different roles that should be dealt with using two potentially different scheduling mechanisms.

2. The design and implementation of a preemptive scheduler is inherently more difficult than a non-preemptive schedulers.

In this chapter, we present an admission control framework that takes the above challenging aspects into consideration. The presented framework stores some similarities with the framework presented in Chapter 3.

The rest of the chapter is organized as follows. Section 4.2 introduces the networking model, and formulates the admission control problem as a discrete optimization problem. Sections 4.3 describes the main components of the proposed CAC framework. Performance results are then presented in Section 4.4, followed by concluding remarks in Section 4.5.

4.2 System Model and Problem Formulation

A. System Model.

In this chapter, we adopt the same system model described in Chapter 3 (similar to the networking environment considered, for example, in [46] and [81]). For the purpose of making the section self-contained, we present below the main characteristics of the model:

- We consider a cellular network where a fixed number C of dedicated data channels is reserved for serving the multimedia streaming QoS class in a

target cell under analysis.

- The bandwidth offered by each channel is assumed to be sufficient for serving each multimedia connection at a satisfactory data rate, and hence, there is no need for bandwidth adaptation during a connection's lifetime.

- User requests to the base station for receiving multimedia streams are assumed to be served by multimedia servers located either on the Internet, or hosted by the wireless network provider. In both cases, the time delay to fetch a requested stream, and make it available for downlink wireless transmission is assumed to consume a negligible amount of time. Hence, user requests that are admitted for service have their arrival time to the base station coinciding with the arrival time of the associated stream of packets from the multimedia server to the base station.

- We considers the following two types of delays.

  Start of service delay: for streaming requests that receive service, the time elapsed between a user request, and the time the base station starts the downlink transmission.

  Service interruption delay: this type of delay may arise if the base station becomes overloaded serving multiple connections that require more than the allocated number of channels. The arrival of handoff connections is one reason that may give rise to an overload condition.

The above delay constraints present both opportunities, and challenges to managing network resources. On the one hand, the CAC can exploit the tolerable start of service delays to admit more traffic connections. On the other hand, the CAC should admit new connections only if they can be served to completion, along with other possible ongoing connections.

B. Problem Formulation

Similar to the approach taken in Chapter 3, we formalize the CAC design problem as an optimization problem. For the purpose of dealing with the problem as a preemptive scheduling problem, we introduce some new notation. To simplify

the design, we assume that all multimedia streams arrive at the wireless network as sequences of fixed length packets. Each packet is transmitted on the downlink in some unit of time, refereed to also as a *time slot*. The CAC views time as a sequence of such time slots. We refer to the amount of data that can be transmitted in one time slot over one channel as a *service data unit* (SDU). At any instant of time t, the CAC is given the following inputs:

$R(t) = \{r_1, r_2, r_3, ..., r_{nR}\}$: a set of n_R connection requests. Each request is specified by the following:

- ℓ_i: connection length,
- a_i: arrival instant,
- sd_i: maximum allowable start of service delay,
- rd_i: maximum allowable total service interruption delay, and
- w_i: weight.

All intervals and time instants assume integer values in units of the CAC's time slots mentioned above. Request r_i is dropped from the system if it cannot be started on, or before, time $s_i = a_i + sd_i$, or cannot be completed before time $s_i + l_i + rd_i$.

T_{plan} (the planning interval of the CAC's scheduler): when invoked, the CAC admits requests that can be scheduled to start no later than the interval defined by this parameter.

We incorporate the streaming QoS class delays mentioned above into the problem formulation as follows:

1. For a connection r_i originating in the target cell, we set both sd_i and rd_i to the desired values associated with the request.

2. In contrast, if r_i is a handoff request from a neighboring cell to the target cell, then r_i is an ongoing connection that should experience a limited interruption delay. Hence, we set $sd_i = 0$, and rd_i to the remaining amount of its tolerable service interruption delay.

3. Request weights in the problem formulation are intended to give more control on the behavior of the CAC. For example, the weights may be set by the network provider to provide service differentiation among classes of requests. Alternatively, they can be set by a resource management module prior to calling the CAC to give priority to some requests over others.

As done in Chapter 3, we define an ideal (short term) CAC as one that given a set $R(t)$ of already arrived requests, the CAC admits a subset of requests $R' \subseteq R(t)$ that maximizes the objective function $\sum_{r_i \in R'} w_i$ such that all admitted requests can be scheduled to start no later than T_{plan} units of time, and all new, and currently ongoing, connections can finish without violating their respective delay constraints. Connection requests in R' are then forwarded to the link-layer packet scheduler, together with the computed start of service times. Under ideal user channel conditions, all streams in R' can then be transmitted to their respective users without violating the delay constraints.

C. Background Results and Remarks

The above problem of designing an ideal CAC calls for selecting a set of connections of maximum total weight that can be preemptively scheduled, along with the currently ongoing connections, so as to satisfy the required delay constraints. Similar to a remark made in Chapter 3, this type of preemptive scheduling problems is related to the class of interval scheduling problems for which various related results appear in the literature. Below, we highlight some background results related to the above combinatorial scheduling problem:

1. When the number of available channels $C = 1$, the non-preemptive and preemptive scheduling problems are identical. We recall from Chapter 3 that when $C = 1$, and T_{plan} is unconstrained, the above problem has been shown to be NP-complete.

2. The problem is also NP-complete when $C > 1$, and all deadlines are equal (cf. problem [SS12] on preemptive scheduling in [26]). Approximation algorithms with bounded approximation ratios for non-preemptive variants of the above problem are given in [5].

3. When $C = 1$, and T_{plan}, and the delay intervals assume bounded values, the problem can be solved in polynomial time by a dynamic programming algorithm (see, for example, [11], [23] and the references therein).

4. For restricted problem instances with zero start of service delays (T_{plan}, and service interruption delays can assume arbitrary values), the feasibility problem that asks whether any given subset of connections can be scheduled over any given number of channels can be solved efficiently using a reduction to network flows (see, e.g., [55]).

D. An Integer Linear Program Formulation

We now present an integer linear program (ILP) for solving the above optimization problem for any given set $R(t)$ with n_R connection requests, and number of channels C. The formulation uses the following remarks, variables, and notation:

- We recall that all intervals (e.g., connection duration,s and delays) are assumed to be multiples of the time slot mentioned above. Denote by t_{max} the maximum possible number of time slots such that any served connection completes service in some time slot $t \in [0, t_{max}]$.

- For a request r_i that arrives at time a_i, we denote by $slot(a_i)$ the time slot that starts at instant a_i.

- For a request r_i that ends no later than time $e_i = a_i + sd_i + l_i + rd_i$, we denote by $slot(e_i)$ the time slot that ends at instant e_i.

- For a request r_i, we use a binary variable x_i where $x_i = 1$ means request i is selected for service, and $x_i = 0$ means request i is not selected for service.

- For a request r_i, and time slot $t, t \in [slot(a_i), slot(e_i)]$, we use a binary variable $x_{i,t}$ where $x_{i,t} = 1$ means that request i is served in time slot t. Otherwise (if $x_{i,t} = 0$) then request i is not served in time slot t

Due to the computational complexity of the scheduling problems, no numerical results are obtained from the above formulation.

Maximize

$$\sum_{i=1}^{n_R} w_i.x_i$$

Subject to:

for each time slot $t \in [0, t_{max}]$:

$$\sum_{i=1}^{n_R} x_{i,t} \leq C,$$

for each connection $i \in [1, n_R]$:

$$\sum_{t=slot(a_i)}^{slot(e_i)} x_{i,t} = l_i.x_i,$$

for each possible connection $i \in [1, n_R]$: $x_i \in \{0, 1\}$,

for each connection i and slot $t \in [slot(a_i), slot(e_i)]$: $x_{i,t} \in \{0, 1\}$.

Figure 4.2: Integer Linear Program Formulation

4.3 Framework Architecture

In this section we adopt a CAC architecture similar to the one devised in Chapter 3 where the CAC has two main components, called the T_{plan} adapter, and the scheduler, as described below.

4.3.1 The T_{plan} Adapter

Earlier, we remarked that the length of the T_{plan} parameter affects both the amount of work done by the CAC's scheduler, and the set of admitted requests. In this section we give a numerical example to show that longer T_{plan} intervals don't necessarily result in better admission decisions. Hence, the need to develop a method for tuning its value.

Example.

In Figure 4.3, assume that $C = 2$ channels are available for serving all seven equally weighted requests. Assuming $T_{plan} = 2$, the CAC module is invoked when free channels are available at instants $t = 0$ and 7:

- At $t = 0$, the input $R(t) = \{r_1, r_2, r_3, r_4\}$. The optimum solution is $R =$

$R(t)$, since all four requests can be scheduled over two channels such that no connection starts later than T_{plan} (here, both of r_3 and r_4 are scheduled to start at $T_{plan} = 2$). Hence, all four connections are admitted.

- At $t = 7$, connections r_5, r_6, and r_7 are no longer available since $sd = 2$ for each request.

So, only 4 requests are served by the network. Now, assume $T_{plan} = 1$. The CAC module is invoked when free channels are available at instants $t = 0$, 2, and 5:

- At $t = 0$, the input $R(t) = \{r_1, r_2, r_3, r_4\}$. The optimum solution is $R = \{r_1, r_2\}$, since only these two requests can start no later than $T_{plan} = 1$.
- At $t = 2$, the input $R(t) = \{r_3, r_4, r_5, r_6, r_7\}$. An optimum solution is $R = \{r_5, r_6, r_7\}$.
- Finally, at $t = 5$, the input is $R(t) = \{r_3, r_4\}$.

The optimum solution is $R = R(t)$. And all seven requests are served by the network. □

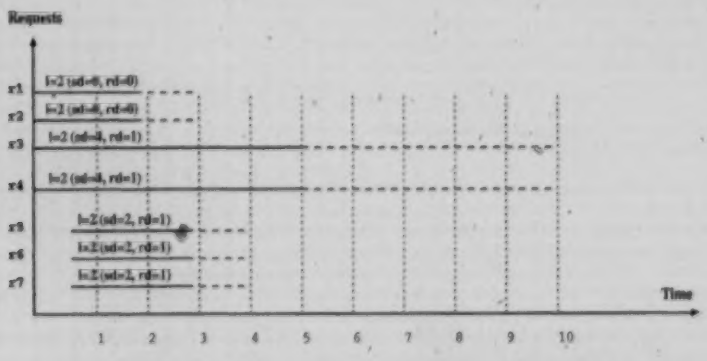

Figure 4.3: A Scenario with Seven Connection Requests

The above example suggests that the use of a short T_{plan} interval may be advantageous when there is plenty of incoming requests to the system such that the dropped requests are replaced quickly with new arrivals. On the other hand, the use of relatively long planning intervals, and the computation of more elaborate preemptive schedules to identify which requests to admit may be advantageous when

the incoming traffic requests is bursty, with average burst inter-arrival time close to T_{plan}.

A. Adapting T_{plan}

The method introduced in Chapter 3 for the T_{plan} adapter is used here. This method can be summarized as follows:

At some time instant t, the scheduler is assumed to be invoked with two, or more, connection requests awaiting service, and at least one free channel. The adapter computes the time difference $\triangle_{SLC}$ since the scheduler has last been invoked. At time t, the current value of T_{plan} is called expired if $T_{plan} > \triangle_{SLC}$. The following cases are then distinguished.

1. No arrival since last scheduler invocation, and T_{plan} has not expired: the current value of T_{plan} is viewed as adequate.

2. One, or more, new arrivals occurred since last scheduler invocation, and T_{plan} has not expired: the arrival of new requests require invoking the scheduler; the current value of T_{plan} is perceived to be longer than needed. To shorten the value, we use an exponential moving average: $T_{plan} = \alpha T_{plan} + (1-\alpha)\triangle_{SLC}$ (we use $\alpha = 0.5$ in the simulation study).

3. T_{plan} expired: the current value of T_{plan} is perceived to be inadequately short; we extend T_{plan} gradually, up to a maximum value of the longest tolerable delay of requests awaiting service.

4.3.2 The Scheduler

As mentioned above, when invoked at time t, the CAC's scheduler takes as input:

1. the number of free channels at time t, denoted $C(t)$,

2. a set $R(t) = \{r_1, r_2, ..., r_{nR}\}$ of outstanding unexpired connection requests, each request r_i is described by its duration l_i, and the (possibly updated) values of its weight w_i, and delay constraints (denoted sd_i and rd_i), and

3. the current value of the T_{plan} parameter.

The scheduler's objective is to select a subset $R' \subseteq R(t)$ of maximum total weight $\sum_{r_i \in R'} w_i$ that can be scheduled preemptively on the $C(t)$ channels, so that no connection starts later than $t + T_{plan}$. The selected subset R' is admitted for service, and the CAC informs the link-layer packet scheduler with the computed start of service times.

In light of the NP-completeness result mentioned earlier, it is unlikely that an efficient exact algorithm exists for solving any arbitrary instance of the problem. Our design relies on using a heuristic algorithm, as outlined below.

As done in Chapter 3, the scheduling algorithm devised here utilizes three components:

1. An outer loop part, similar to Figure 3.4, that calls suitable one-channel, or two-channel packing algorithm for scheduling a subset R' of the input requests so that all selected requests can be scheduled to start within T_{plan} units of time. The outer loop starts with an empty solution set: $R' = \phi$, and incrementally adds new connections to R'. The incremental additions to R' are done by pairing the available channels into $\lfloor C(t)/2 \rfloor$ pairs, and repeatedly invoking a function called preemptive 2-channel packing scheduler (P2-CPS) presented below.

2. A function called preemptive 2-channel packing scheduler (P2-CPS), that takes as input $C = 2$ channels, the information associated with a set of traffic requests, and the computed T_{plan} value, and selects a subset of requests that can be preemptively scheduled so that all selected requests can be scheduled to start within P2-CPS time units.

3. A function to solve the above scheduling problem for $C = 1$ channels. The function is invoked with the unpaired channel if $C(t)$ is odd integer.

A. Two Channel Packing Scheduler

Figure 4.4 presents a pseudo code for the P2-CPS function. As done in Chapter 3, to simplify the presentation, throughout this section we treat the time instant when the scheduler is invoked as $t = 0$. All traffic requests arrival times, and remaining tolerable delay intervals are relative to this time instant.

Function P2-CPS takes as input the timing information of connection requests in the input set $R(T)$, T_{plan}, and two time instants i, j such that, $i, j \in [0, T_{plan}]$, where the two channels are available for service. The output of the function, denoted R'', is a subset of $R(t)$ that can be scheduled over two channels.

Function P2-CPS:

Input: the timing information of connection requests in the input set $R(t)$, T_{plan}, and two time instants i and j such that, $i, j \in [0, T_{plan}]$, where the two channels are available for service.

Output: a subset R'' of $R(t)$ that can be scheduled over two channels so that each request starts within T_{plan} units.

1. Initialize the solution set $R'' = \phi$;

2. do {

   2.1. Let S be a subset of $R(t) - R''$ of size $|S| \leq 3$ connections satisfying the following conditions:

   (a) If $|S| = 1$ connection then the connection can be scheduled to start at the earliest time instant when one of the two channels is available for service (i.e., the connection starts at $min(i, j)$).

   (b) If $|S| = 2$ or 3 connections then two connections of S can be scheduled to start on instants i, and j respectively.

   (c) S has a maximum possible total weight, subject to the above two conditions.

   2.2. If no such S exists then exit the loop.

   2.3. Else, construct a sub-schedule of S over two channels satisfying conditions (a) and (b) above.

   2.4. Let i and j be the earliest times according to the sub-schedule constructed above when the two channels become available.

   2.5. Add S to R''.

}

3. Return the computed schedule R''

***Figure 4.4:** Pseudo-code for Function P2-CPS.*

The function works iteratively. In each iteration (the body of the do loop) the function selects a subset, denoted S, of at most 3 connection requests that can be scheduled to start when the two channels are available for service (instants i and j in the pseudo code). To decide whether a subset S of 3 connections admit such scheduling, we present and prove below a certain schedulability condition.

The particular subset S chosen in an iteration is required to have the maximum possible total weight among the candidate subsets (cf. condition 2.1(c)). If no such subset S exists, then the function terminates and returns the computed subset R''. If such subset S exists then the function constructs a sub-schedule of the set S in step 2, and determines the two time instants (denoted i and j) where the two channels become available according to the constructed schedule.

An iteration terminates by adding the selected subset S to the current solution set R'' before the start of the next iteration. The following example illustrates the composition of a solution set R'' from the subschedules constructed over all iterations.

Example.

Figure 4.5 illustrates a possible output R'' computed by function P2-CPS. The output is assumed to be computed in three iterations (compositions of three subschedules). Each of the subschedules 1, and 3, has two connections, and subschedule 2 has three connections. □

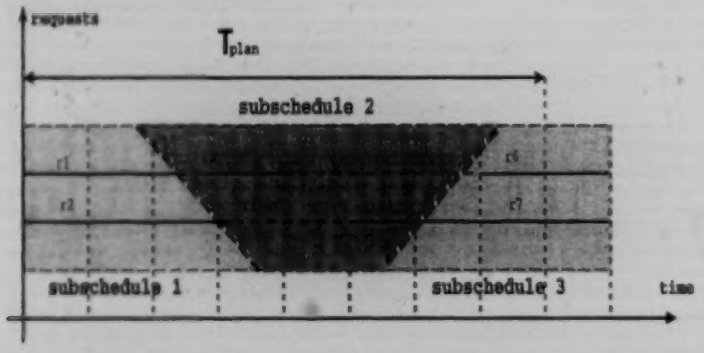

***Figure 4.5:** A View of the Solution Computed by Function P2-CPS as a Composition of Multiple Sub-schedules*

To compute such set S in each iteration, the function operates on the set of all undelayed, and delayed (up to a maximum of T_{plan} units) instances of the currently unadmitted connection requests; this set of all possible connection instances is denoted $DR(t)$, and is formally defined as follows:

$DR(t) = \{r_{i,d} | r_i \in R(t)$, and $d = 0, 1, ..., min(sd_i, T_{plan})$, and $r_{i,d}$ starts at time $t + d\}$.

Searching for the best connections to include in each subschedule is done exhaustively over the set $DR(t)$ of undelayed and delayed instances. Deciding whether any given three instances in $DR(t)$ can be scheduled over two channels is done by exploiting the following observation.

B. Schedulability Condition.

Let r'_1, r'_2, and r'_3 be three instances of connections in the set $DR(t)$, where each instance r'_i, $i = 1, 2, 3$, starts at time t_i, and has an allowable maximum service interruption delay rd_i. Assume, without loss of generality, that $t_1 \leq t_2 \leq t_3$, as shown in Figure 4.6. To test whether all three instances can be served preemptively using only two channels, we note that:

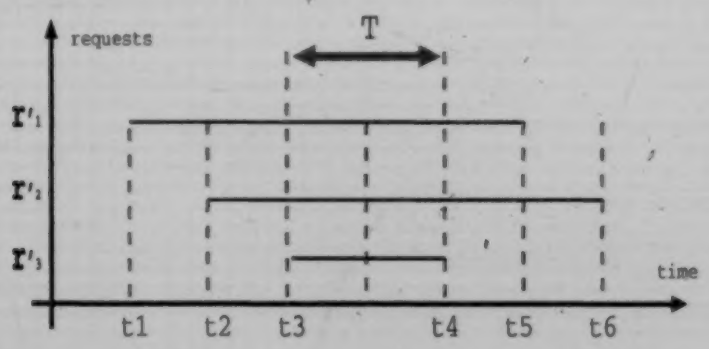

***Figure 4.6:** Three Instances of Requests that can be Served Using two Channels*

Proposition. Let T be the length of the interval during which all three connections overlap, and let $rd_{min} = min(rd_1, rd_2, rd_3)$. The three connections can be served using two channels if $T \leq 2.rd_{min}$.

Proof. Denote by t_4, t_5, and t_6 the end points of the three instances, where $t_4 \leq t_5 \leq t_6$. The connections can then be scheduled as follows. During $[t_1, t_3 + \frac{T}{2}]$ serve r'_1 and r'_2. During $[t_3 + \frac{T}{2}, t_4]$ serve r'_2 and r'_3. During $[t_4, t_4 + \frac{T}{2}]$ serve r'_1 and r'_3. At $t_4 + \frac{T}{2}$, at least one connection finishes service. Serve the remaining connections (if any) until completion using the two available channels. Since each connection is delayed by $\frac{T}{2} \leq rd_{min}$, the schedule is feasible.

Running Time. Let l_{min} denote the length of the shortest connection request. Then function P2-CPS computes at most $\frac{T_{plan}}{l_{min}}$ sub-schedules. To compute each subschedule, the function requires $O(n^3_{DR}(t))$ to exhaustively search the set $DR(t)$ of

all possible undelayed and delayed instants of input requests for a set S of maximum possible total weight with at most 3 requests that satisfies the required conditions. Thus, in the worst case, the function requires $O(\frac{1}{2}(\frac{T_{plan}}{l_{min}})^3)n_{DR})$ time. □

C. Improving the Constructed Schedule.

As mentioned above, the schedule constructed by function P2-CPS may contain sub-schedules where three connection requests are multiplexed over 2 channels. The preemptive schedule of such 3 connections introduces delays (gaps) between successive transmissions of the same connection request.

Media playback at the user end may deteriorate if any such delay period exceeds a certain threshold value; this threshold value depends on the media player allocated buffer size, and possibly the utilized encoding scheme. In this section, we show the following result:

Proposition. Any three connection requests in any such subschedule can be served so that the time delay between any two successive transmission of any connection does not exceed one time slot.

Proof. Consider the timing in Figure 4.6, we show an allocation of time slots to the three connection requests r'_1, r'_2, and r'_3 during the interval $[t_3, t_3 + 3\lceil\frac{T}{2}\rceil]$, which includes the interval $[t_3, t_4 + \frac{T}{2}]$ that satisfies the required condition (i.e., the time delay between any two successive transmission of any connection does not exceed one time slot).

The allocation uses the following procedure:

- Divide the interval $[t_3, t_3 + 3\lceil\frac{T}{2}\rceil]$ into $x = \lceil\frac{T}{2}\rceil$ *cycles*, where each cycle has 3 time slots.
- In each cycle, serve the requests during the available 3 slots as follows: (r'_1, r'_2), (r'_1, r'_3), and (r'_2, r'_3).

In the above procedure observe that:

1. the procedure guarantees that each connection is delayed a maximum of 1 time slot in each scheduling cycle.
2. the maximum total delay encountered by any connection is bounded by $x = \lceil\frac{T}{2}\rceil] \leq rd_{min}$, as required. □

4.4 Performance Results

In this section we obtain simulation results to investigate the following aspects:

- the strength of the devised preemptive scheduling scheme, and
- the effectiveness of the devised admission control scheme that combines the T_{plan} adapter module, and the preemptive schedule.

Similar to the approach taken in Chapter 3, we introduce modifications to the CAC algorithm in [46] to obtain a competitive scheme that is used in our comparative study. We refer to the modified scheme scheme as a *time-slotted predictive* CAC (TS-Predictive CAC, for short). The TS-Predictive CAC enhances the predictive CAC scheme described in Chapter 3 by utilizing a scheduler that works at the time-slot level, as described below.

4.4.1 Time-Slotted Predictive CAC

For the purpose of obtaining a competitive CAC scheme to be used in our performance study, we modify the predictive CAC scheme, described in Section 3.8.2 so that the new scheme employs a *fine-grain* scheduler that works as follows:

- The scheduler works at the CAC service data unit (SDU) level (where an SDU is the amount of data that can be transmitted to the user in one time slot over one channel).
- The scheduler is assumed to be invoked at the beginning of every time slot interval.
- In each time slot, the scheduler determines which SDUs to be forward to the link packet scheduler using a priority based rule. The rule considers the connection requests in ascending order of the ratio of the remaining connection's duration time (part of ℓ_i) to the remaining allowable service interruption time (part of rd_i).

Time-Slotted Predictive CAC:

Input: traffic, and mobility parameters, as described above, as well as CAC control parameters T and P_{qos}

1. **The CAC queue:**
   - Upon arrival of a handoff connection, admit the connection by adding it to the scheduler's queue;
   - Upon arrival of a connection originating from the same cell, perform the overload test as described in [46] and accept if $P_{CO} \leq P_{qos}$, otherwise delay the connection (if not expired).
2. **CAC's scheduler queue:**
   Use a priority based rule to select a subset of SDUs for forwarding to the link layer packet schedule.
3. remove expired connections periodically

Figure 4.7: *Pseudo-code of the Time-slotted Predictive CAC Scheme*

The CAC module then forwards as many as C (the number of available channels) SDUs from connections with higher assigned priorities to the link level packet scheduler. Figure 4.7 presents the architecture of the TS-Predictive CAC.

Discussion. We remark that the use of the fine-grain slot-by-slot scheduler alone without the cell overload prediction mechanism has a good potential of delivering competitive results. To explore this aspect further, we evaluated the achieved throughput of the CAC in Figure 4.7 with, and without, the cell overload prediction mechanism. The results in Figure 4.7 show a slight improvement in the effective throughput in the case of utilizing the cell overload prediction mechanism. The results in the rest of this section are evaluated using the scheme with cell overload prediction in effect.

4.4.2 Simulation Environment

Table 4.1 summarizes the parameters used in the simulation. The system capacity C is set to 200 channels, serving a connection request is done by allocating 10 channels at each time. Thus, the system can support up to 20 connections at the time. Also, $1/\mu = 200$ seconds, $1/\delta = 100$ seconds, $T = 20$ seconds, and $P_{qos} = 0.2$. For each connection request r_i, we set $d_i = 0.1\ell_i$. The arrival rate λ is varied

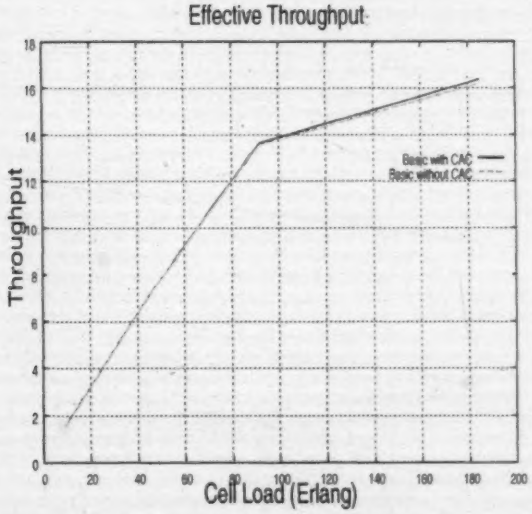

Figure 4.8: *TS Predictive with and without CAC scheme*

to generate different levels of offered traffic loads.

Table 4.1: *Simulation Parameters*

Parameter	Value
C	200 channels
$1/\mu$	200 seconds
$1/\delta$	100 seconds
T	20 seconds
P_{qos}	0.2
d_i	$0.1\ell_i$

For user mobility, we assume that new connection arrivals form a Poisson process with arrival rate denoted λ. A connection stays in the cell for a period of time that follows an exponential distribution. Call duration is assumed to follow exponential distribution as well. User i remain in the same cell with probability P_r, and with probability P_h it hands off to a neighbor cell. The time a user spends in a cell is called residence time.

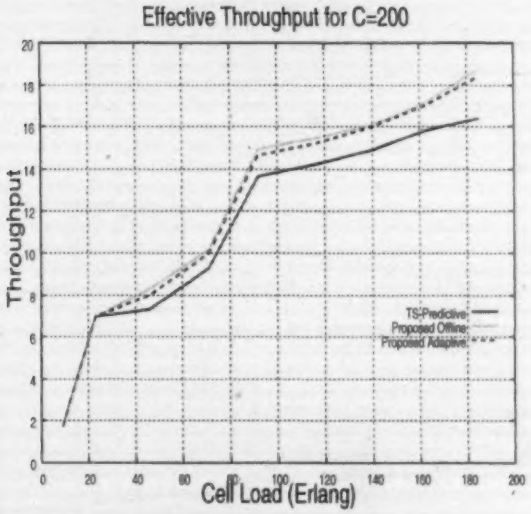

Figure 4.9: Effective Throughput

4.4.3 Numerical Results

In this section the curves labeled "Offline" refers to the results obtained by using the devised framework on trace files with all connection requests information stored in the files. Thus, the devised CAC algorithm has full knowledge of all requests in such curves. For convenience, we refer to such curves as being produced by the "offline algorithm". On the other hand, the curves labeled "Adaptive" refers to the results obtained by using the devised framework with no such knowledge. For convenience, also, we refer to such results as being obtained by the "adaptive algorithm". Thus, the throughput results obtained by the offline algorithm is expected to act as an upper bound on the throughput results obtained by the adaptive algorithm.

1. Effective Throughput

The effective throughput results are depicted in Figure 4.9. The obtained results show that both the adaptive, and the offline algorithms achieve higher throughput than the TS-predictive algorithm. Increasing the offered traffic load, increases the achieved throughput for all schemes. As the main objective function for the adaptive as well as the offline schemes is to maximize the cell effective throughput, the obtained results show that the design goal is served.

2. Percentages of Forcibly Terminated Connections

The forcibly terminated connections refer to both blocked handoff, and ongoing dropped connections. The results on the percentage of forced termination connections are depicted in Figure 4.10. We remark the following:

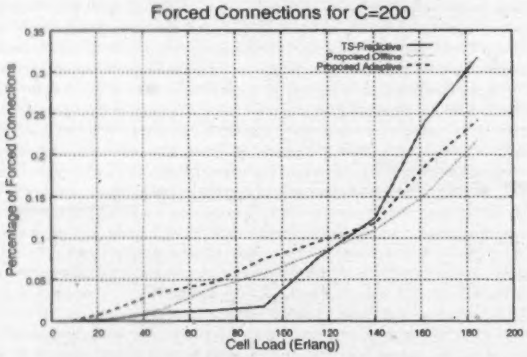

Figure 4.10: *Forced Terminations Percentage*

- At low cell load, all schemes achieve similar results in terms of forcibly terminated connections.
- At medium loads, the TS-predictive CAC algorithm outperforms both adaptive and offline schemes.
- At high system load the TS-predictive CAC algorithm forcibly terminated more connection than the other two schemes.
- Since the TS-predictive scheme accepts handoff requests without subjecting it to the overload test, forced terminations in this scheme occurs as a result of terminating an ongoing newly generated, and handoff connections. at low and medium loads, the system rarely terminates any ongoing connections, In such cases, the percentage of forced terminations is low. At high cell load
- For both the adaptive and the offline schemes, forced terminations may occur in two cases: as a result of rejecting handoff connection requests, or as a result of terminating ongoing handoff connections. Both schemes, achieved good results at low cell load. As the load increases, the forcibly terminated

connections increase gradually. Figure 4.10 shows a smooth increase in the forcibly terminated percentage as opposed to the TS-predictive scheme.

- For all cell loads, the offline scheme provides an upper bound on the forced terminations achievable by the adaptive algorithm.

3. Percentages of Completed Connections

The results of the percentages of completed connections are shown in Figure 4.11. The results show that the TS-predictive scheme serves more connection requests to completion than other schemes proposed in this chapter. As the TS-predictive scheme does not restrict the admission condition of newly generated connections in the cell to connections that maximizes the system throughput (the scheme only attempts to avoid overload conditions), it is expected that the system admits connections with low weight (e.g., short durations). On the other hand, since the CAC scheme proposed in this chapter accepting connections that maximize the system throughput, the scheme favors connections with high weights (e.g., long duration). Such difference in the type of the accepted connections by both schemes is the key to explain the results. We also remark that the offline curve acts as an upper bound on the adaptive curve.

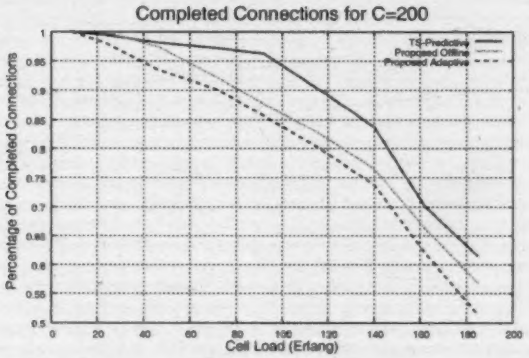

***Figure 4.11:** Completed Connections.*

4. Percentages of Blocked Connections

The results of the percentages of blocked connections are shown in Figure 4.12. Blocked connections refer to rejected connections originated in the target cell. Sim-

ilar to the results obtained for the percentage of completed connections, the TS-predictive scheme outperforms the proposed scheme. This results can be explained using the same argument (type of accepted connections) used to explain the results for the percentage of completed connections.

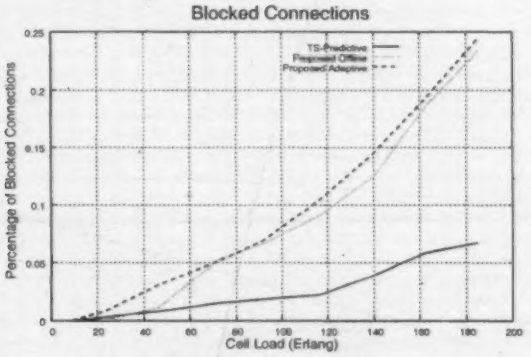

Figure 4.12: *Blocked Connections.*

5. Remarks.

Although the TS-predictive CAC algorithm achieved higher percentage of completed connections, it falls behind in achieving higher throughput than the proposed schemes. One possible reason for this behavior is due to the type of connections being admitted by each algorithm. The TS-predictive algorithm tends to choose short duration connections. In the other hand, the other two algorithms tends to admit connections with higher weights (e.g., longer durations).

4.5 Concluding Remarks

In this chapter we investigate the design of a CAC for managing multimedia connection requests in a wireless cellular network where the streaming QoS service class is allocated a fixed number of dedicated data channels. The proposed design considers two different types of delay bounds for each request. The architecture combines the use of an adaptive approach to deduce a suitable planning interval for deciding which connections to process with a scheduling heuristic. The reported results indicate the usefulness of the proposed approach.

Chapter 5

Admission Control of Delay Bounded Traffic for CDMA Networks

In the previous chapters we investigated the design of call admission control schemes for serving delay bounded traffic in wireless networks where a fixed number of channels is assumed to be allocated to serving the traffic.

Our main contributions in this chapter are on investigating the problem in wireless cellular networks that utilize a discrete sequence code division multiple access (DS-CDMA) air interface. In such networks, the available data transmission capacity is affected by the multi-user access interference (MAI). The cell may undergo overload conditions if the total transmission power required to maintain service to all ongoing connections exceeds the total base station transmission power allocated to serve the traffic.

To achieve acceptable performance, the CAC framework devised in this chapter has two main components:

- A scheme that takes into account mobility of users with ongoing connections to predict the cell overload probability after a prescribed prediction interval. The scheme admits a traffic stream only if the estimated cell overload probability after a prescribed prediction interval does not exceed a specified threshold value.

- A scheduling module that works at the CAC service data unit (SDU) that takes service interruption delays into account and decides which data units to be served.

Performance of the proposed call admission control framework is analyzed using simulation. A preliminary version of the work in this chapter appears in [43].

5.1 Introduction

In this chapter, we consider the design of a CAC framework for serving delay bounded traffic in networks that utilize the CDMA architecture, as described, for example in [24, 30]. In the context of serving connections of the streaming QoS class, the presence of the multi-user access interference (MAI) is known to complicate the CAC design in at least the following two aspects:

- Streaming connections require relatively high rates. Increasing a base station's transmission power to support high data rates increases the intra-cell and inter-cell multiple access interference (MAI). MAI has the effect of decreasing the available cell bandwidth, as well as increasing the possibility of running into cell overload conditions.

- Provisioning streaming services in a dependable way requires maintaining session's quality (e.g. the allocated data rate) during both intra-cell, and inter-cell user movements. Intra-cell user movements cause significant changes in the channel path loss between a mobile user and the serving base station while the user is possibly receiving a streaming flow. Thus, the base station has to continually monitor the ongoing streaming connections, and adjust the transmission power accordingly. Therefore, a successful CAC scheme has to be aware of user mobility.

The approach taken in this chapter devises a CAC framework that utilizes the following modules:

1. A predictive CAC scheme that utilizes a priori knowledge of inter-cell, and intra-cell user mobility distributions to predict the cell overload probability. The computed cell overload probability, and the allowable start of service delays are used to decide whether to accept, delay, or reject incoming connection requests.

2. A CAC scheduler that works at the CAC time slot level, and considers the service interruption delays to determine in each time slot which CAC service data units (SDUs) should be forwarded to the network's packet scheduler for further transmission to the end users.

We note that the method used to compute the cell overload probability used in the first module is not specific to serving delay bounded traffic (see, e.g., [21]).[1]

The architecture devised in this chapter, and in [21], extends the predictive CAC architecture in [46] to the case of CDMA networks. For prior work, we also notice that the work of [75] uses a control theoretic approach for designing a CAC that dynamically adjusts the capacity of guard channels in CDMA cellular networks so as to maintain the handoff dropping rate at a target level. Simulation of voice dominated traffic is used to analyze performance. We note that the above approaches do not address the problem of serving the admitted flows at some data rate during the entire session time while taking mobility into account, as done in this work.

For cellular networks with fixed cell capacity, the work of [46] devises a combined CAC and adaptive bandwidth allocation scheme for serving streaming connections. The results of [46] deal with an adaptive multimedia networking framework where the bandwidth of an ongoing multimedia connection can by dynamically adjusted several times during the connection's lifetime. That is, the assigned data rates can be decreased (when overload conditions arise), or increased (when some connections terminate) dynamically. The devised CAC algorithm works by

[1]The main ideas underlying the design of the first module (the predictive CAC) has been developed by the author in the present context of serving delay bounded traffic, and concurrently in [21] for the purpose of designing an adaptive bandwidth allocation framework for serving streaming connections. The results obtained in this chapter are based on an independent implementation of the following modules: the predictive CAC module, the CAC's scheduler module, and the overall event driven simulation program.

predicting the state of a cell (the number of active users, including handoff users) after some prediction time interval in the future. The CAC devised in [21], and utilized in this chapter, generalizes the approach used in [46] to the more complex case of predicting the state of the cell in a CDMA environment.

The rest of the chapter is organized as the following: Section 5.2 presents the system model for the proposed architecture. Section 5.3 present the call admission control scheme. Section 5.4 presents an algorithm for overload probability computation. Section 5.5 presents the scheduling module. Section 5.6 presents simulation environment and the numerical results. Finally, Section 5.7 presents the chapter conclusion.

5.2 System Model

Throughout the chapter, we consider a UMTS-like cellular network architecture (see, e.g., [30]) operating in the *frequency division duplex* (FDD) mode where each user streaming flow is served by a dedicated data channel. The newly added functionality of the proposed CAC and scheduling modules are assumed to be embedded in the radio network control (RNC) part. Multimedia streaming requests are assumed to be served by servers located either on the Internet, or hosted by the public land mobile network (PLMN) provider, and attached to the gateway GPRS support node (GGSN) of the core network (CN) unit.

We assume that packets of each streaming connection experience most of the delays in the wireless network as a result of, e.g., queuing delays from the serving base station to the mobile user. As mentioned above, user mobility is assumed to be the primary reason for causing the required transmission power to exceed the total base station transmission power allocated for serving the streaming QoS class, and consequently the occurrence of queueing delays.

The multimedia server may be capable of delivering a requested multimedia clip at any one of a possible number of available resolutions. In our context, the method used to encode a user requested clip, denoted c_i, determines the time duration required by the base station to transmit the clip (the duration l_i defined below)

at a given data rate (denoted R below) to the end user, and the maximum allowable service interruption delay (denoted d_i below). Furthermore, we utilize the mathematical model introduced in Section 1.3, to determine whether a given collection of users can be served at the given data rate R.

Throughout this chapter, we make the following notations and assumptions:

1. The total base station transmission power allocated to serve connections of the streaming QoS class is denoted P_{total}. Power is assigned to users based on the data transmission rate R (mentioned below), and user distances from the base station.

2. $T_{predict}$: a CAC's parameter that denotes a length of the prediction period used in computing the cell overload probability from the mobility transition diagram (introduced later).

3. R: a fixed data rate used for downlink transmission of a streaming connection's data over a transport channel.

4. Each user may have maximum of one connection active at any given time.

5. t_{slot}: time is slotted. The CAC views time as a sequence of slots, each of length t_{slot} (e.g. 100 msec). As mentioned in Chapters 3 and 4, we remark that the value of t_{slot} is chosen to be a multiple integer of the packet scheduler slot length. We also remark that t_{slot} should not be set to a large value that allows the average large scale path loss to a user to change significantly (due to user mobility) during a single slot.

6. CAC's SDU: at each t_{slot} each active user may receive only one CAC service data unit (SDU) from the serving base station. All SDUs have the same maximum amount of data ($t_{slot} \times R$). The base station may decide to delay or drop some SDUs in case of overload.

7. α_{qos}: A CAC's parameter ($\alpha_{qos} > 0$) used to set an upper bound on the maximum acceptable processing (queuing + transmission) time experienced

by any of the admitted connections. To explain the role of α_{qos}, we introduce the following scheduling parameters for a connection with index i:

(a) a_i: (arrival time) the slot number at the beginning of which the first SDU of the connection can be transmitted,

(b) l_i: (connection's length) the number of slots required to transmit all encoded bits in the stream to the intended user at the distinguished rate R,

(c) c_i: (completion time): the last slot number used in serving the connection. Thus, the total processing time of the connection at the base station is given by $1 + c_i - l_i$ slots, and

(d) d_i: an upper bound on the acceptable total processing time (in slot units) at the base station (i.e., an upper bound on the $1+c_i-l_i$ quantity), given by

$$d_i = \lceil (1 + \alpha_{qos}) l_i \rceil \tag{5.1}$$

8. P_{admit}: a CAC's parameter (a probability) used in accepting new connections.

A few remarks about the above assumptions follow below. We assume that a mobile device requesting and receiving streaming connections is limited with respect to the energy and memory resources; hence, the choice of using a single serving data rate R to simplify communications. In addition, due to the limited buffering capacity available to the mobile for compensating delay jitter during playback, frequent delays of user packets result in an unacceptable playback performance. If a connection i does not receive the required i slots of service within the acceptable delay interval $[a_i, a_i + d_i - 1]$ then that connection is considered to be inadequately served. We further assume that such badly served connection is forcibly terminated by the base station without charging the user.

To account for the impact of such service termination policy on the network provider's revenue, we use the average effective throughput as one of the main performance measures; here, the average is taken over all traffic streams that have been

successfully served to completion within the desired delay limits (that is, excluding the streams that have been forcibly terminated).

5.3 Call Admission Control Architecture

In this section, we start by introducing some concepts that allow us to construct a CAC that takes into account both inter-cell and intra-cell user mobility to determine whether it is suitable to accept a new arrived connection request.

The concepts presented below describe how a cell can be partitioned into a number of regions with different path loss characteristics, and how user mobility among such regions can be described, and how to approach the problem of estimating the probability that a cell will likely be in an overload condition after some predefined interval in the future.

- **Cell Decomposition.** Due to the MAI in CDMA cells, we note that, it is insufficient to take the number of active streaming users in the cell as the only measure of cell load. Rather it is preferred to consider a more detailed account of the base station transmission power requirements of different classes of users. In this regard, the approach taken in [21] is based on classifying the active streaming users according to the amount of base station transmission power required to provide service at a specified data rate. The classification is based on the average large scale path loss experienced by each user. Here, we follow the same approach.

  In particular, in [21], the CAC scheme assumes that each cell is partitioned into a fixed number r ($r \geq 2$) of regions, called rings. Each ring corresponds to a geographical area of the cell where the average large scale wireless path losses from the serving base station to each point of the ring fall within some predetermined range (e.g., [70 dB, 110 dB]). Different rings correspond to non overlapping regions with distinct path loss bounds, as illustrated in Figure 5.1. In general, dividing the cell into a large number of rings results in a more accurate model at the expense of increased computational requirements of the CAC scheme.

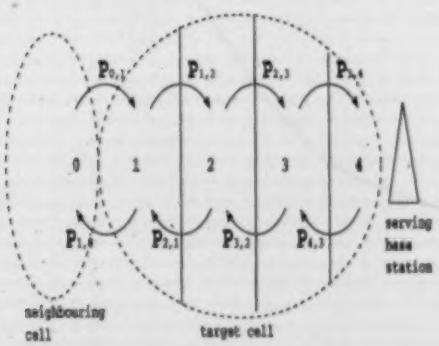

Figure 5.1: Ring-based Mobility Model

- **Cell States.** The state of the cell at any instant t is captured by an occupancy distribution vector $N = (n_0, n_1, \ldots, n_r)$, where n_0 is the number of the streaming class users that may perform handoff to the target cell, and for $i \in [1, r]$, n_i is the number of active ring-i users in the target cell.

- **Cell Overload Probability.** An occupancy distribution vector N is called an *overload* distribution if the base station cannot allocate sufficient transmission power to serve each mobile at the distinguished rate R, assuming that each mobile experiences the maximum observed average large scale path loss in its respective ring. The set of all overload distributions is denoted OL.

  Furthermore, given that the cell is in state N at sometime t, the overload probability, denoted $P_{N,OL}$, is defined as the probability that the cell will be in some overload state at time $t + t_{predict}$.

- **Traffic and Mobility Models.** User traffic and mobility in the cell (during an operation period of interest) is modeled using the following distributions.

  - The arrival of streaming connection requests to the target cell is assumed to be Poisson with rate λ connections per second.
  - The time duration of each connection is exponentially distributed with average $1/\mu$ seconds.
  - For connections initiated in ring $i \in [1, r]$ of the target cell, we refer to the time duration the connection is served in that particular ring as ring-i

residence time. After that time duration, the user may depart to another ring. The scheme assumes that ring-i residence time is exponentially distributed with average $1/\delta_i$.

- Handoff traffic from neighboring cells to the target cell is modeled using a similar setup: that is, the handoff traffic is viewed as coming from a new ring (denoted ring-0 in Figure 5.1) lying outside the target cell. The scheme also assumes that ring-0 residence time is exponentially distributed with average $1/\delta_0$ (δ_0 is the handoff rate from the neighboring cells to the target cell).

- **Transition Probabilities.** User movement across rings during a certain epoch of the day is modeled by a transition diagram where rings and transitions are represented by nodes and directed edges, respectively. Here, $p_{i,j}, i \neq j, i, j \in [0, r]$, denotes the probability that an active user moves from ring i to ring j after spending the residence time mentioned above. The transition probabilities out of any ring $i \in [1, r]$ satisfies $\sum_{i \neq j} p_{i,j} = 1$. For the special case of $i = 0$ (i.e., for ring-0 representing the neighboring cells), we have $\sum_{j=1}^{r} p_{0,j} \leq 1$.

- **Ring Residence and Departure Probabilities.** The probability that a test connection initiated in ring i at time t will continue to be served in the same ring during the prescribed prediction interval $t_{predict}$ is denoted $p_{r,i}$; thus, $p_{r,i} = e^{-(\mu+\delta_i)t_{predict}}$. In addition, the probability that a test connection initiated in ring i will depart to another ring during the next $t_{predict}$ time units is denoted $p_{d,i}$ ($p_{d,i} = 1 - e^{-\delta_i t_{predict}}$). Thus, corresponding to each (i, j) transition in the mobility diagram, the product $p_{d,i}\, p_{i,j}$ is the probability that a test connection initiated in ring i will depart to ring j during the next $t_{predict}$ time units.

5.3.1 Physical Layer Constraints

The CAC scheme devised in [21] combines a SINR-based admission control scheme (see, e.g., [38]) with a mechanism to predict the state of the target cell after a

prescribed prediction interval $t_{predict}$. The main physical layer constraints that the scheme applies are described below.

Given an occupancy distribution N associated with a set $M = \{1, 2, ..., m\}$ of users in the target cell (i.e., $m = \sum_{i=1}^{r} n_i$), determining whether N is an overload distribution requires that the RNC estimates the minimum amount of base station transmission power that should be allocated to serve each user. We denote such vector of required transmission power levels by $P_{tx} = (P_{tx,1}, P_{tx,2}, \ldots, P_{tx,m})$. Determining P_{tx} (if one exists) is carried out by solving a linear system of equations where the equation associated with mobile receiver $u \in M$ is described in detail in Section 1.3.

5.3.2 Admission Procedure

Given the above framework, the the admission procedure works as follows:

1. Upon arrival of a new streaming connection request, the CAC computes the occupancy distribution vector N that results if the newly arrived request is admitted. The RNC then tests whether there exists a feasible power allocation P_{total} to serve all in-cell connections of the distribution N at data rate R.

2. If there is no feasible allocation, and the new arrived request can be delayed, the base station delays the request, else, the base station rejects the request.

3. Else (if a feasible solution exists), the procedure described in the next section is executed to decide whether the following condition is satisfied:

$$P_{N,OL} \leq P_{admit} \tag{5.2}$$

where P_{admit} is a QoS admission control probability for the streaming class set by the network provider. Finally, the RNC admits the request if condition 5.2 is satisfied.

5.4 Overload Probability Computation

The work of [21] presents a method for computing the overload probabilities (the $P_{N,OL}$ quantities used in equation 5.2), and using such values in the devised CAC framework. The same method can be used in step 3 of the CAC mentioned above. The following is a summary of important observations made in [21] related to the above aspect.

1. Given a transition diagram G, and a non-overload occupancy distribution vector $N = (n_0, n_1, \ldots, n_r)$ at some time instant t, the work of [21] presents a brute force algorithm for computing the probability $P_{N,OL}$. The brute force algorithm works in time $O(\prod_{i \in r} n_i^{|E_i^+|})$, where $|E_i^+|$ is the number of outgoing arcs in the diagram. We note that, each node corresponding to an in-cell ring has a self transition that represents ongoing connections that continue in the same ring during the interval $[t, t + t_{predict}]$.

2. The CAC devised in [21] uses the procedure mentioned above to compute the set of all $(R, t_{predict})$-*safe* (called safe states for short, and denoted $\mathcal{S}$) occupancy distributions: a distribution N is safe if it is not an overload distribution, and satisfies the admission condition $P_{N,OL} \leq P_{admit}$ after $t_{predict}$ units of time. The computations of $\mathcal{S}$ are assumed to be done offline.

3. The CAC architecture of [21] presents a method of computing a reduced representation of $\mathcal{S}$ that can be stored in a suitable data structure (e.g., a B+ tree), that can be searched in $O(r)$ time (i.e., for a fixed number of rings r, the search requires a constant time).

4. Given an occupancy distribution N, the CAC checks the condition of the inequality 5.2 in real time by searching the above data structure to check whether $N \in \mathcal{S}$.

We remark that the brute force algorithm mentioned in point 1 above is most useful when the transition diagram G has a few number of rings, and the maximum total number of active connections that can exist simultaneously in the system is

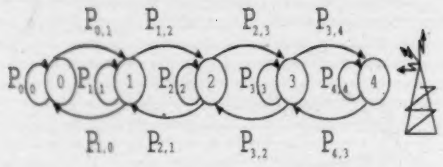

Figure 5.2: *Two Way Symmetric Mobility Diagram*

small. The fast growth rate of the running time encourages further research to identify restricted classes of transition diagrams that are useful in practice, and enable more efficient algorithms.

In this section, we present one such restricted class of mobility transition diagrams called the *two-way symmetric* mobility diagram, see for example Figure 5.2. In such diagram, each ring i, $i \geq 1$, has at most two outgoing transitions to the adjacent rings $i-1$, and $i+1$, in addition to the self-loop transition (i, i).

Two-way symmetric transition diagrams are useful, for example, in modeling slowly moving users in a highly congested area, such as users leaving a sport event, where all users are served by the same base station. In such context, users requesting short (few minutes) multimedia streams are likely to receive such stream while traversing a short distance.

In next section we present an algorithm that can be used as a tool for analyzing the overload probability for two-way symmetric mobility diagrams.

5.4.1 Two-way Symmetric Diagram Overload Probability Calculations

We start by presenting the following remarks, and definitions:

1. **Support Vectors**: given an overload ring occupancy distribution vector

$$N' = (n'_0, n'_1, \ldots, n'_r)$$

the *support* vector of N', denoted $support(N')$ is a partially specified vector:

$$(*, \ldots, *, l_i, n'_{i+1}, \ldots, n'_r), i \in [1, r]$$

satisfying the following conditions:

(a) the index $i \in [1, r]$ (i.e., the index $i = 0$ of the ring representing the neighboring cells in the model is not a possible value of the index i in the definition).

(b) $1 \leq l_i \leq n'_i$.

(c) the index $i \in [1, r]$, is the smallest index such that the vector

$$(0, \ldots, 0, l_i - 1, n'_{i+1}, n'_{i+2}, \ldots, n'_r)$$

is feasible (i.e., not an overloaded vector), but the vector

$$(0, \ldots, 0, l_i, n'_{i+1}, n'_{i+2}, \ldots, n'_r)$$

is an overload vector. □

Example 1. Suppose a cell is divided into $r = 3$ rings: (r_1, r_2, r_3). Suppose that $N' = (2, 3, 2, 3)$ is an overload vector. Suppose that scanning N' from right to left has revealed that the vector $(0, l_1 = 2, n'_2 = 2, n'_3 = 3)$ is an overloaded vector, but the vector $(0, l_1 - 1 = 1, n'_2 = 2, n'_3 = 3)$ is not an overload vector. Moreover, the index $i = 1$ is the smallest index for which the above required property holds. Then $support(N') = (*, 2, 2, 3)$.

2. **Closure of support vector**: given a support vector

$$R = (*, *, \ldots, *, l_i, n'_{i+1}, \ldots, n'_r)$$

of some overload occupancy distribution vector, the closure of R, denoted $closure(R)$, is the set of all vectors where each vector

$$Q = (q_0, q_1, \ldots, q_i, q_{i+1}, \ldots, q_r)$$

satisfies the following conditions:

- For each possible index $j \in [i + 1, r]$: $q_j = n'_j$ (i.e., the two vectors Q and R agree on the elements from $i + 1$ to r.
- For the index i: $q_i \geq l_i$.
- For each possible index $j \in [0, i - 1]$: $q_j \geq 0$.

Equivalently, $closure(R)$ is the set of all vectors of the form

$$Q = (q_0, q_1, \ldots, q_{i-1}, l_i + \epsilon_i, n'_{i+1}, \ldots, n'_r)$$

, where each possible $q_j \geq 0$, and $\epsilon_i \geq 0$. □

Example 2. In Example 1, $R = (*, 2, 2, 3)$ is the computed $support(N')$ vector, where $N' = (2, 3, 2, 3)$. Now, assume that cell has at most $N = 8$ possible streaming class users (note that in N', at most $n_0 = 2$ users lie outside the cell). Then,

$$closure(R) = \begin{array}{l}\{(0,2,2,3),(1,2,2,3),(2,2,2,3),\\ (0,3,2,3),(1,3,2,3),(2,3,2,3)\}\end{array} \square$$

The problem considered in this section can be stated as follows: given a non-overload ring occupancy distribution vector $N = (n_0, n_1, \ldots, n_r)$ at time t, and an overload distribution vector $N' = (n'_0, n'_1, \ldots, n'_r)$ at time $t + t_{predict}$, compute the probability $P_{N,closure(support(N'))}$ that the vector N results in an overload vector in the $closure(support(N'))$ at time $t + t_{predict}$. We remark that if OL is the set of all overload vectors, then the set OL can be partitioned into disjoint sets, where each set is the closure of some support vector.

Thus, given a non-overload vector N at time t, an algorithm to solve the above problem can be used to compute the overload probability $P_{N,OL}$. Table 5.1 summarizes the symbols and notations used by both the admission and the overload calculation procedures.

Table 5.1 summarizes the symbols and notations used by both the admission and the overload calculation procedures.

5.4.2 Two-way Symmetric Diagram Overload Probability Algorithm

Figure 5.5 presents the main algorithm described in this section. To present the algorithm we need the following definitions:

- **Definition** (Augmentation vector of a 2-way symmetric mobility diagram): Given a ring occupancy distribution vector $N = (n_0, n_1, \ldots, n_r)$, an aug-

Table 5.1: Important Admission and Overload Calculation Procedures Parameters

Parameter	Definition
$P_{N,OL}$	Probability that distribution N results in an overload distribution at the end of the interval $[t, t + t_{predict}]$.
P_{admit}	A threshold on the overload probability for accepting/rejecting new requests.
OL	The set of all overload ring occupancy distribution vectors.
$N = (n_0, n_1, \ldots, n_r)$	A cell occupancy distribution vector considered by the CAC.
$support(N')$	For a given overload vector N', $support(N')$ is a partially specified vector, as described above.
$closure(R)$	For a given partially specified vector R, $closure(R)$ is a set of derived vectors, as described above.

mentation vector of N, denoted

$$N^+ = (N_i^+, \ldots, N_r^+)$$

is a vector satisfying the following conditions:

- the index i of the first element is in the range $[1, r]$, and
- any possible N_j^+ in the augmentation vector N^+ is a pair of numbers

$$N_j^+ = (n_{j,j}, n_{j,j-1})$$

that correspond to a possible distribution of users that may move along the arcs (j, j), and $(j, j-1)$, respectively (see the thick solid arcs in Figure 5.3). Thus, any such pair of numbers satisfies

$$n_{j,j} + n_{j,j-1} \leq n_j$$

(i.e., the sum of the users moving along the two specified arcs $\leq n_j$).

Example 3. If $N = (2, 3, 1, 3)$ and $N^+ = (N_1^+, N_2^+, N_3^+)$ where,

$$N_1^+ = (n_{1,1} = 1, n_{1,0} = 0),$$
$$N_2^+ = (n_{2,2} = 0, n_{2,1} = 0), \textit{and}$$
$$N_3^+ = (n_{3,3} = 2, n_{3,2} = 1)$$

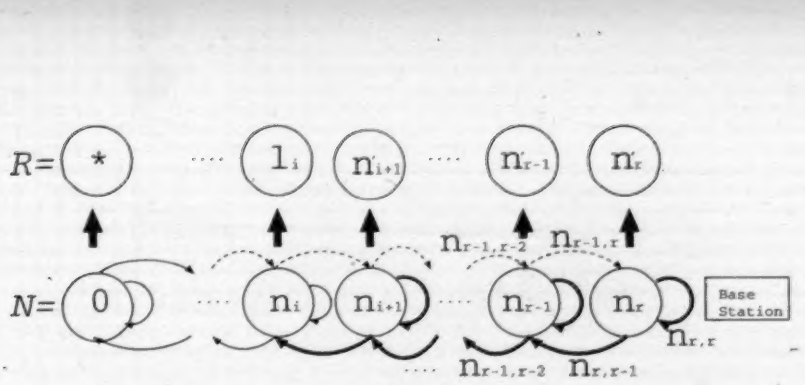

Figure 5.3: *Two-way Symmetric Overload Calculation Diagram*

then N^+ is a valid augmentation vector. □

Note that, for any possible pair N_j^+ (corresponding to ring-j with n_j users) in an augmentation vector, the numbers $n_{j,j}$ and $n_{j,j-1}$ don't specify how many users out of the n_j users may have moved to a ring closer to the base station (see the dashed lines in Figure 5.3), nor how many have terminated service.

- **Definition** ((N, R)-Augmentation vector of a 2-way symmetric mobility diagram): Given a ring occupancy distribution vector $N = (n_0, n_1, \ldots, n_r)$, and a support vector

$$R = (*, *, \ldots, *, l_i, n'_{i+1}, \ldots, n'_r)$$

of some possible overload vector, an augmentation vector $N^+ = (N_i^+, \ldots, N_r^+)$ of N is called (N, R)-augmentation vector if starting from N (at time t) we can reach an overload distribution in $closure(R)$ (at time $t + t_{predict}$) by a scenario where users move according to the distribution in N^+. □

Example 4. In Example 3, we note that if $N = (2, 3, 1, 3)$ then

$$N^+ = \{N_1^+ = (1, 0), N_2^+ = (0, 0), N_3^+ = (2, 1)\}$$

is a valid augmentation vector. Now, suppose that $R = (*, 2, 2, 3)$ is a support vector of some overload vector. We show that N^+ is a valid (N, R)-augmentation vector (see Figure 5.4). To this end, we only need to observe that the following system of linear equations has a solution. In particular, note that:

- Setting $n_{2,3} = 1$ (on a dashed line of Figure 5.4) satisfies the constraint: $n'_3 = n_{2,3} + n_{3,3}$. Note that, in this scenario the number of terminated connections in ring $r_3 = 0$.

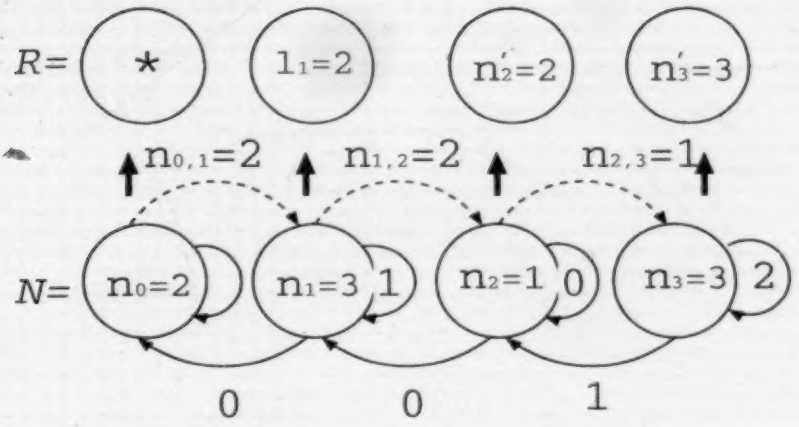

***Figure 5.4:** Augmentation Vector Scenario with 4 Rings*

- Setting $n_{1,2} = 2$ satisfies the constraint: $n'_2 = n_{1,2} + n_{2,2} + n_{3,2}$. Note that, in this scenario the number of terminated connections in ring $r_2 = 1$.

- Setting $n_{0,1} \geq 1$ satisfies the constraint: $l_1 \leq n_{1,1} + n_{0,1} + n_{2,1}$. Note that, in this scenario the number of terminated connections in ring $r_1 = 2$. □

Figure 5.5 presents the overload calculation function. First, the function computes the support vector R of the given overload vector N'. In step 3, the function iterates over all possible (N, R)-augmentation vectors N^+. For each such vector N^+, the corresponding iteration starts by computing the distribution of users moving along the (dashed) lines:

$$n_{i-1,i},\ n_{i,i+1},\ \ldots,\ n_{r-1,r},$$

so that starting from N (at time t) and following the transition vector N^+ and the above computed values, we reach the distribution $(l_i, n'_{i+1}, \ldots, n'_r)$ in rings $(i, i+1, \ldots, r)$, respectively (at time $t + t_{predict}$).

Thus, the computed values together with the values in the augmentation vector N^+ specifies a scenario of users moving along the following arcs:

1. all outgoing arcs incident to rings $i, i+1, \ldots, r$, and

2. the arc $(i-1, i)$.

Using the above values (i.e., the computed values together with the values in the augmentation vector N^+), the function computes the following probabilities:

1. For ring r: the probability $P_{N_r^+|n_r}$ that the following event occurs during the interval $t + t_{predict}$:

   $n_{r,r-1}$ ongoing connections move to ring $r-1$,
   $n_{r,r}$ ongoing connections continue service in ring r, and
   $n_r - (n_{r,r-1} + n_{r,r})$ connections terminate service,

   given that n_r connections are active at time t. This probability is computed by a multinomial distribution.

2. For each ring j, $j \in [i, r-1]$: the probability $P_{N_j^+|n_j}$ that the following event occurs during the interval $t + t_{predict}$:

   $n_{j,j-1}$ ongoing connections move to ring $j-1$,
   $n_{j,j+1}$ ongoing connections move to ring $j+1$,
   $n_{j,j}$ ongoing connections continue service in ring j, and
   $n_j - (n_{j,j-1} + n_{j,j+1} + n_{j,j})$ connections terminate service,

   given that n_j connections are active at time t. This probability is computed by a Multinomial distribution.

3. For ring $i-1$: the probability $P_{N_{i-1}^+|n_{i-1}}$ that the following event occurs during the interval $t + t_{predict}$:

   at least $n_{i-1,i}$ ongoing connections move from ring $i-1$ to ring ring i.

   given that n_{i-1} connections are active at time t. This probability is computed as a sum of Binomial terms.

Running Time. To evaluate the running time, we note that if $N = (n_0, n_1, \ldots, n_r)$ is the input non-overload occupancy distribution vector, and

$$R = (*, *, \ldots, *, l_i, n'_{i+1}, \ldots, n'_r)$$

Function: Overload Calculation

Input: a non-overload vector $N = (n_0, n_1, \ldots, n_r)$, and an overload vector $N' = (n'_0, n'_1, \ldots, n'_r)$.

Output: $P_{N,closure(support(N'))}$: probability that the distribution N results in an overload distribution in the class $closure(support(N'))$ of overload distributions after $t_{predict}$ time units.

1. Compute the support vector $R = support(N') = (*, *, \ldots, *, l_i, n'_{i+1}, \ldots, n'_r)$.

2. $sum = 0$.

3. For each $((N, R)$-augmentation vector $N^+ = (N_i^+ = (n_{i,i}, n_{i,i-1}), \ldots, N_r^+ = (n_{r,r}, n_{r,r-1}))$ {

   - Compute the distribution of users moving along the (dashed) lines:

   $$n_{i-1,i},\ n_{i,i+1},\ \ldots,\ n_{r-1,r}$$

   so that starting from N (at time t), and following the transition vector N^+ and the above computed values, we reach the distribution $(l_i, n'_{i+1}, \ldots, n'_r)$ in rings $(i, i+1, \ldots, r)$, respectively (at time $t + t_{predict}$).

   - For ring r (closest ring to the base station) compute:

   $$P_{N_r^+|n_r} = Multinomial \quad (n_{r,r-1}, P_{depart,r}P_{r,r-1}, n_{r,r}, P_{resident,r}, n_r - (n_{r,r-1} + n_{r,r}), P_{term}).$$

   - For each possible ring $j \in [i, r-1]$ compute:

   $$P_{N_j^+|n_j} = Multinomial \quad (n_{j,j-1}, P_{depart,j}P_{j,j-1}, n_{j,j+1}, P_{depart,j}P_{j,j+1}, n_{j,j}, P_{resident,j}, n_j - (n_{j,j-1} + n_{j,j+1} + n_{j,j}), P_{term}).$$

   - For ring $i-1$ compute:

   $$P_{N_{i-1}^+|n_{i-1}} = \sum_{\alpha \geq n_{i-1,i}}^{n_{i-1}} Binomial \quad (\alpha, P_{depart,i-1}P_{i-1,i}, n_{i-1} - \alpha, 1 - (P_{depart,i-1}P_{i-1,i}).$$

   - Compute the cumulative probability:

   $$sum = sum + \prod_{j=i-1}^{r} P_{N_j^+|N}$$

}

4. Return sum.

***Figure 5.5:** Pseudo Code for Overload Calculation Function*

is the computed support vector then there are at most $O(\prod_{j\in[i,r]} n_j^2)$ augmentation vectors considered in step 3. Computing the $P_{N_j^+|n_j}$ probability in step 3 for each augmentation vector N^+ can be done in a constant time. Thus the overall function requires $O(\prod_{j\in[i,r]} n_j^2)$. □

5.5 Scheduling Algorithm

As mentioned above, the CAC utilizes a scheduling module that works at the CAC time slot level. The scheduling mechanism considers the remaining service interruption delay of each admitted connection to determine which CAC service data units (SDUs) should be forwarded to the network's packet scheduler for further transmission to the end users.

Given the above framework, one way of defining an ideal scheduler is as follows: given a set of n already admitted connections $\{1, 2, ..., n\}$, the average radio path loss to each respective user, the remaining required service time of each connection l_i' (slots), and the remaining allowable service interruption delay time d_i' (slots), $d_i' > l_i'$, we ask for a scheduler that minimizes the number of forced terminations. A few remarks about the problem of finding an ideal scheduler now follow.

1. An optimal scheduler is necessarily preemptive; that is, the scheduler may assign non-adjacent slots for serving a connection (however, no connection is permitted to be preempted within any single slot). For example, if the base station can serve at most two users in any slot, and the system has three identical connections with $l' = 2$, and $d' = 3$ then an optimal scheduler achieving zero forced termination is only possible by preempting some connection for one slot.

2. If the target cell has a fixed capacity, and each connection requires the same amount of the base station transmission power during each scheduling slot (which is not always the case for CDMA cells), then the feasibility problem that asks whether all admitted connections can be scheduled with no forced

termination can be solved efficiently using a reduction to network flows (see, e.g., [55]).

In light of the computational intractability of many scheduling problems, we investigate the performance of a simple priority based scheduling algorithm (see, e.g., [41] for an overview of the use of some priority rules in scheduling). Figure 5.6 presents the scheduling algorithm. The proposed algorithm attempts to minimize the number of forced terminations by assigning higher scheduling priorities to connections with low $\frac{d_i'}{l_i'}$ ratios. In case of the network is unable to support all ongoing connections, the scheduler decides on which connection(s) to be delayed.

Algorithm: CAC's Scheduler:

Input: for each ongoing connection i, the remaining service duration (l_i') and the remaining allowable service interruption delay (d_i').

Output: list of connections to be served in the current t_{slot}.

1. Use equations 1.1 to 1.6 to calculate the total required transmission power for serving all admitted connections, denoted $P_{required}$ for the current t_{slot}.
2. While ($P_{required} > P_{total}$) {
   - 2.1. Calculate the weight $w_i = \frac{d_i'}{l_i'}$ for each admitted connection i.
   - 2.2. Select the connection with highest weight to be delayed.
   - 2.3. Subtract the delayed connection required power from $P_{required}$.

}

***Figure 5.6:** Scheduling Algorithm Description*

5.6 Performance Results

In this section we obtain simulation results to investigate the following performance measures of the devised CAC framework: percentage of completed connections, percentage of blocked connections, percentage of forcibly terminated connections, and effective throughput.

The achieved results are compared against the results obtained by utilizing three other CAC schemes referred to as: the *number*-based CAC scheme, the *delay*-based CAC scheme, and the *power*-based CAC scheme, as described below.

5.6.1 Description of CAC Schemes Used for Comparison

A number of proposed admission control schemes for serving both data and voice in CDMA networks are based on utilizing the instantaneous available information in the cell [35, 3, 80, 48]. In this section, we present three such schemes, of which the first two schemes (the number-based, and the power-based) have been discussed in the literature, but they don't exploit the allowable delays of the traffic connections, and the third scheme (the delay-based scheme) is a new scheme discussed in this thesis, and aims at exploiting the existing allowable delays of the traffic.

A. The Number-Based CAC Scheme.

This scheme uses a threshold value, denoted N_{max}, that specifies the maximum number of connections that can be active at any instant. The main steps performed by this scheme are shown in Figure 5.7. The value of N_{max} is static and may be computed according to various measures (e.g., based on the average path loss, or the maximum average path loss experienced by the connections in the cell). In this section, we consider two sets of results obtained by this scheme. The first set utilizes $N_{max} = 30$, we refer to this results as number-30. And, the second set utilizes $N_{max} = 15$, and we refer to this results as number-15.

Algorithm: Number-based CAC:
Input: Number of ongoing connections.
1. If (number of active connections $< N_{max}$) then
   accept the connection request
2. Else, reject the connection request

***Figure 5.7:** The Number-Based CAC Algorithm*

B. The Power-Based CAC Scheme.

In this scheme, the algorithm considers the instantaneous cell power information to decide whether to accept, reject, or delay the new connection request. The main steps performed by this scheme are shown in Figure 5.8.

In this scheme, the CAC admits a new connection, if and only if, the base station has enough transmission power to support the ongoing connections, and the

Algorithm: Power-based CAC:
Input: current average path loss experienced by ongoing and new connections.

1. Estimate the base station transmission power required to serve the ongoing connections and the new request, denoted P_{all}.
2. If ($P_{required} + P_{all} \leq P_{total}$) then
   accept the connection request
3. Else, reject the connection request

Figure 5.8: *The Power-Based CAC Algorithm*

new connection request. The scheme estimates all connection (both ongoing and new) path loss values according to their current distances from the base station. This scheme considers the instantaneous path loss effect; it doesn't exploit any information regarding the connection mobility distribution, or QoS delay requirements.

C. The Delay-Based CAC Scheme.

The delay-based CAC aims at achieving improved performance over the two previously mentioned CAC schemes (i.e., the number-based, and the power-based schemes) by taking into account the following factors:

- the remaining allowable service interruption delay of the ongoing connections, and the new arrived connection, and
- the variability of the number of SDUs that can be served concurrently in any time slot during a prediction interval $[t, t+t_{predict}]$. The variability arises since users associated with the ongoing connections, and the new arrived connection are assumed to be moving (according to known trajectories) during the prediction interval.

We use the following example to illustrate the basic idea of the delay-based CAC.

Example. In Figure 5.9, we assume that:

1. The interval $[t, t + t_{predict}]$ contains three time slots.

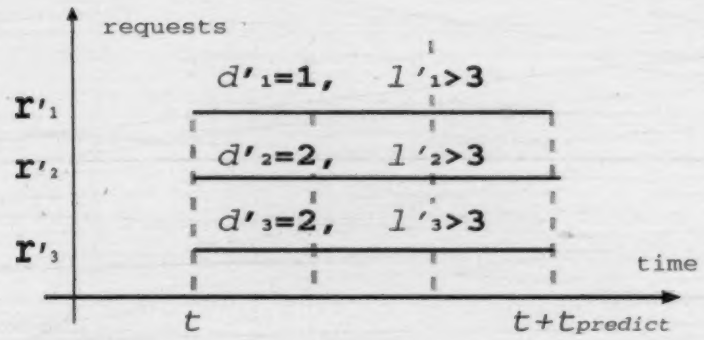

Figure 5.9: *Example of 3 Requests to Illustrate the Delay-based Scheme*

2. r_1 and r_2 are ongoing connections, and r_3 is a new arrived connection, where $(d'_1 = 1, l'_1 > 3)$, $(d'_2 = 2, l'_2 > 3)$, and $(d'_3 = 2, l'_3 > 3)$, and the corresponding users are moving during the interval $[t, t + t_{predict}]$.

3. The predicted channel path loss in each time slot does not allow the base station to serve all three connections in any time slot.

4. To determine how many (and which) connections can be served during each time slot, we execute the basic steps of Figure 5.6 in each time slot. The obtained results from the computation are assumed to allow serving one SDU in the first time slot, two SDUs in the second time slot, and one SDU in the third time slot. Thus, the base station predicts that $1+2+1 = 4$ SDUs can be served during the prediction interval, and thus the number of unserved SDUs $= 9 - 4 = 5$. We denote such number by F_{extra}.

5. To determine whether the new arrived connection can be accepted, the CAC compares F_{extra} with the sum $F_{delay} = \sum_{i=1}^{3} d'_i = 1+2+2 = 5$. If $F_{delay} \geq F_{extra}$ the CAC accepts the new connection. □

Figure 5.9 presents the basic steps of the CAC using the following variables:

1. F_{extra}: the number of unserved SDUs during the interval $[t, t + t_{predict}]$.

2. $F_{delay} = \sum d'_i$, where the sum is over all ongoing connections, and the new arrived connection.

Algorithm:Delay-based CAC:

Input: For each ongoing, and the new arrived connections, specify: l_i' = the remaining duration, and d_i' = the remaining allowable service interruption delay .

1. Set $F_{extra} = 0$.

2. For each time slot in the prediction interval $[t, t + t_{predict}]$ {

   2.1. Calculate the base station transmission power required for serving the ongoing connections, and the new arrived connection.

   2.2. Using the main steps as in Figure 5.6, compute the number of SDUs that can't be served in the current time slot, denoted F'.

   2.3. $F_{extra} = F_{extra} + F'$.

}

3. Compute the aggregate remaining service interruption delay for the ongoing connections, and the new arrived connection F_{delay}.

4. If $(F_{extra} \leq F_{delay})$ then accept the new connection. Else, reject.

Figure 5.10: *The Delay-Based CAC Algorithm Description*

5.6.2 Simulation Parameters

Table 5.2 summarizes the parameters used in the simulation study. The parameters are classified into the following categories:

- Mobility parameters: we use a random mobility model when simulating the performance of the the number-based, delay-based, and power-based CAC schemes. In this model, users move with fixed speed of 1 meter/second until a user reaches its new destination, after reaching a destination, a user is assumed to pause for an exponentially distributed interval, denoted T_{pause} with average of 100 seconds.

  In contrast, we use the ring-based mobility model described in Section 1.5 in simulating the performance of the predictive CAC scheme described in Section 5.3. In this model, the ring residence time, denoted $T_{ringresidence}$, is exponentially distributed with average of 100 seconds, and a handoff probability $P_h = 0.3$.

- Traffic parameters: connection duration $T_{connection}$ is randomly generated from an exponential distribution with average 200 seconds. On the other hand, connection inter-arrival time (i.e., the time between the end of one connection, and the start of a new connection request from the same user) is varied to generate different system loads. Connections are served with a fixed data rate $R = 128$ kbps.

- Physical layer parameters: the 19-cell model is used for power and interference calculations, (c.f. Equations 1.1 to 1.6). The physical layer parameters are described in Section 1.3, and their numerical values are listed in Table 5.2.

5.6.3 Numerical Results

The obtained results are discussed below.

1. Effective Throughput

Table 5.2: Parameters and Associated Values

Physical Layer Parameters			
P_{total}	25 Watt	Path loss at d_0	142 dB
SNR	7 dB	n	4
W	4.096 Mcps	Standard Deviation	4
α (orthogonality factor)	0.2	Cell radius	1000 meters
d_0(reference distance)	711 meter	Number of users	30
η_0(noise density) $3.98 \times 10^{-18} mWatt/Hz$			
User Mobility Parameters			
Speed	1 meter/second	$T_{ringresidence}$	100 seconds
T_{pause}	100 seconds	P_h	0.3
Traffic Parameters		**CAC Parameters**	
Data rate R	128 kbps	α_{qos}	10%
T_{slot}	0.1 seconds	$T_{predict}$	20 seconds
$T_{connection}$	200 seconds	P_{admit}	0.1

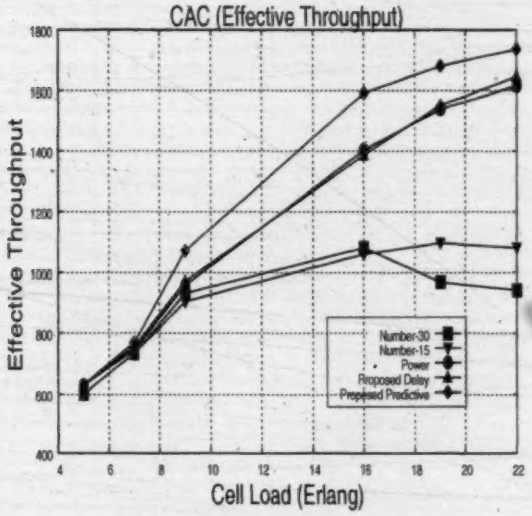

***Figure 5.11:** Effective Throughput*

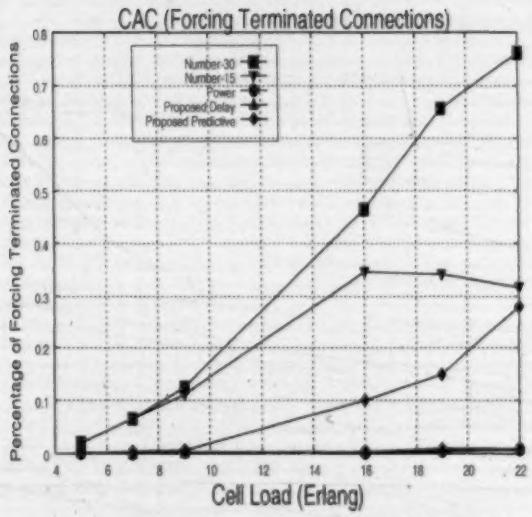

Figure 5.12: *Forced Terminated Connections*

Figure 5.11 depicts the effective throughput results. The predictive scheme achieves the highest effective throughput. The effective throughput increases as the load increases for all the schemes except for the number-30 scheme. The delay-based and the power-based schemes achieve a close and competitive results.

2. Percentages of Forced Terminated Connections

Percentages of forced terminated connections results are depicted in Figure 5.12. The power-based and the delay-based schemes outperform the other schemes, as they impose a very tight restrictions on the admitted connections. Again, the predictive scheme achieves a reasonable performance relative to the number-based scheme. We note that the number-15 scheme forced termination ratio is stable at high traffic load. On the other hand, the number-30 forced termination ratio increases dramatically as the system load increases.

The use of the CAC's scheduler in the predictive scheme enabled the CAC framework to maintain the percentage of the forcibly terminated connections under 30%. The delay-based scheme utilizes a similar approach as to the one used by the CAC's scheduler; the use of this technique restricts the number of accepted connections by the delay-based scheme and thus it achieves negligible percentage of forced termination.

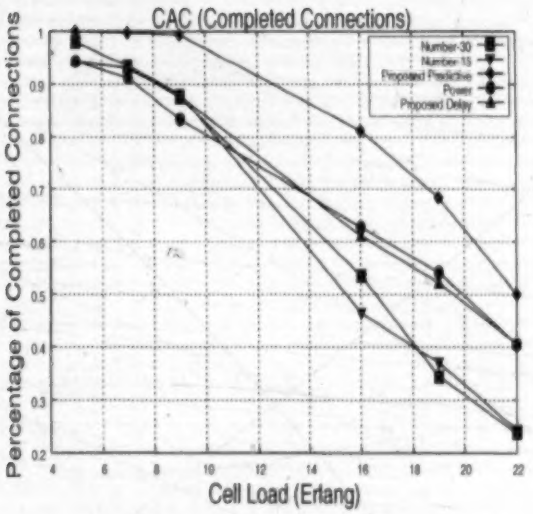

Figure 5.13: Completed Connections

3. Percentages of Completed Connections

Percentages of completed connections results are depicted in Figure 5.13. The obtained results show that the predictive scheme outperforms the other schemes in terms of the percentage of completed connections, while the number-based scheme achieves the worst performance. At light load, all schemes show competitive performance. As the load increases, the predictive scheme outperforms the other schemes. Both power-based and delay-based scheme show competitive performance. The results obtained here have a direct impact on the achieved effective throughput. As more connections are being served to completion, the overall network throughput increases.

4. Percentages of Blocked Connections

Percentages of blocked connections results are depicted in Figure 5.14. In this set of results, as the maximum offered load at any given time does not exceed 30 requests, the number-30 scheme achieves 0% blocking. The adaptive scheme manages to achieve a good performance relative to the other two schemes. Again, the power-based and delay-based schemes achieve a close performance.

5. Remarks.

One can make the following observations:

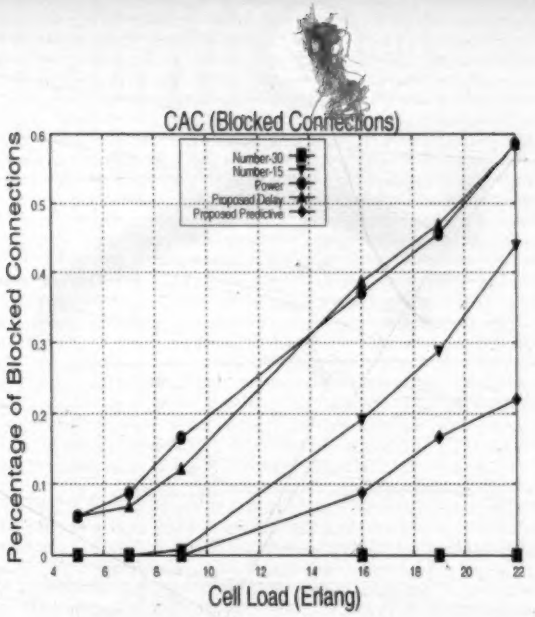

***Figure 5.14:** Blocked Connections*

1. The predictive-based scheme achieves the best results in terms of the system effective throughput and completed connections. It also achieves a reasonably good results in terms of the blocked and forcibly terminated connections.

2. The number-30 scheme accepts all incoming connections (i.e., it doesn't block any request). This approach proved to be inefficient as it achieves very poor results. Increasing the system load results in decreasing the effective throughput in case of number-30 scheme.

3. The delay-based and power-based schemes are very selective, and accepts a new request, if and only if, the system can support the new connection. Although, both schemes blocks the highest number of connections, they achieve very close and competitive results in terms of effective throughput and completed connections. Both schemes, achieve the best results in terms of forcibly terminated connections.

4. Although predictive-based CAC blocks more connections than the number-30 scheme, it still manages to balance the ratio of blocked to forced terminated connections to achieve better performance in terms of increasing the number of completed connections and effective throughput. Increasing effective

throughput increases the service provider profit. And, increasing the number of completed connections increases users satisfaction.

5.7 Concluding Remarks

In this chapter, we devise a call admission control framework that consists of an admission procedure and a scheduling module. The admission procedure considers users mobility in order to estimate the system overload probability. The scheduling module utilizes the individual request tolerable delays to reduce the system overall forced terminations. The obtained numerical results show that predicting the system state yields better performance in terms of increasing the system throughput and the number of completed connections.

Chapter 6

Conclusion and Future Work

6.1 Summary

The growing interest in extending the Internet streaming services to mobile users in modern cellular networks has led to many research efforts on designing call admission control, and scheduling mechanisms to best serve such delay sensitive traffic.

In this thesis, we formalized a class of problems of serving such traffic by considering bounds on two types of delays associated with each connection request: a start of service delay that a mobile user may tolerate before being served, and a service interruption delay that corresponds to the acceptable aggregate delay that may be incurred during the reception process. Overload conditions in the wireless network are assumed to be the prime reason of data transmission delays on the downlink from the base station to mobile users.

In Chapters 3 and 4, we devised call admission control schemes for serving streaming connections under delay constraints of the above types in networks where a fixed number of channels are assumed to be available for serving the incoming requests. In Chapter 5, we devised a call admission control scheme that works in CDMA networks where multi-user interference affects the available bandwidth for serving end users. The design of the proposed CAC schemes utilizes a number of analytical results that are derived in the thesis. The obtained simulation results indicate improvement over the other competing strategies.

In particular, for networks with fixed number of channels, we devise CAC mechanisms that keep track of network state at any instant by utilizing a scheduling

mechanisms that take into account the delay constraints of individual traffic connection requests. Two types of scheduling mechanisms are considered in the thesis for the above purpose: a non-preemptive mechanism that assumes that a connection request is served to completion without service interruption, and a preemptive mechanism that aims at achieving higher throughput by allowing service preemption. In each case, the thesis

- develops framework for the devised CAC and the underlying scheduling mechanism,
- presents quantitative analysis of the designed schedulers, and
- evaluates the performance of the devised framework by simulation.

A novel contribution of the thesis is the design and analysis of CAC architecture of the above type to serve delay bounded traffic.

Moreover, for CDMA networks, we devise a CAC mechanism that keeps track of network state at any instant by keeping track of both intra-cell and inter-cell mobility of served users in order to estimate the cell overload probability after a prediction interval in the future. Such CAC architecture has been devised in the literature for rate sensitive (but not particularly delay bounded) traffic. A novel aspects of the thesis is in extending the architecture to our present context of serving delay bounded traffic in soft capacity networks.

6.2 Future Work

We conclude by identifying some possible research directions related to serving delay sensitive traffic through packet-switched domains in cellular wireless networks. Similar to the problems considered in the thesis, the directions proposed below calls for designing suitable admission control schemes, and/or packet scheduling schemes, with matching mathematical models to examine their performance.

The first direction considers networking environments where the network provider defines a set $B = \{R_1, R_2, \ldots, R_n\}$ of data rates for serving users of the streaming

class. So, the network offer n different service classes to the users. Each user subscribes to one of the available data rates. A user subscribing to rate R_i can enjoy receiving a multimedia stream with this rate if his/her wireless condition permits (else, the rate is decreased). We then seek to formalize the problem of admitting and serving the incoming connections so as to optimize an objective function that may be defined as minimizing the number of terminated connection requests, or maximizing the achieved throughput subject to constraints similar to the constraints dealt with in the thesis.

The second direction also considers the availability of a set $B = \{R_1, R_2, \ldots, R_n\}$ of data rates for serving the streaming class. Here, all subscribers belong to only one service class. The network can increase the transmission rate to an active user if his/her wireless condition allows such increase, and an increase in the data rate results in satisfying the delay constraints of the admitted connections.

Problems in the above two directions benefit from the continuing advances in wireless networks, and generalize the problems described in the thesis. Hence, they appear to be worthwhile pursuing.

Bibliography

[1] A. S. Acampora and M. Naghshineh. An architecture and methodology for mobile-executed handoff in cellular ATM networks. *IEEE Journal on Selected Areas in Communications*, 12(8):1365–1375, October 1994.

[2] R. Akl and A. Parvez. Global versus local call admission control in CDMA cellular networks. In *CITS 04: Communications. Information and Control Systems, Technologies and Applications*, pages 283–288, 2004.

[3] I. Akyildiz, D. Levine, and I. Joe. A slotted CDMA protocol with BER scheduling for wireless multimedia networks. *IEEE Transactions on Networking*, 7(2):146–158, April 1999.

[4] M. Atallah. *Algorithms and Theory of Computation Handbook*. CRC, 1998.

[5] A. Bar-Noy, S. Guha, J. Naor, and B. Schieber. Approximating the throughput of multiple machines in real-time scheduling. *SIAM Journal on Computing*, 31(2):331–352, 2001.

[6] A. Bashandy, E. Chong, and A. Ghafoor. Network modeling and jitter control for multimedia communication over broadband network. In *INFOCOM 1999*, volume 2, pages 559–566, March 1999.

[7] J. C. R. Bennett and H. Zhang. WF2Q: Worst-case fair weighted fair queueing. In *IEEE INFOCOM 96*, pages 120–128, March 1996.

[8] C. Bettstetter. Smooth is better than sharp: a random mobility model for simulation of wireless networks. In *MSWIM '01: Proceedings of the 4th ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, pages 19–27, New York, NY, USA, 2001. ACM Press.

[9] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for Ad Hoc network research. *Wireless Communications and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2(5):483–502, 2002.

[10] Y. Cao and V. Li. Scheduling algorithms in broad-band wireless networks. *Proceedings of the IEEE*, 89:76–86, January 2001.

[11] M. Carlisle and E. Lloyd. On the k-coloring of intervals. *Discrete Applied Mathematics*, 59:225–235, 1995.

[12] H. Che, S. Kumar, and C. Kuo. QoS-aware radio resource management scheme for CDMA cellular networks based on dynamic interference guard margin (IGM). *Computer Networks*, 46:867–879, 2004.

[13] B. Chen, A. Vliet, and G. Woeginger. An optimal algorithm for preemptive on-line scheduling. *Operations Research Letters*, 18:127–131, 1995.

[14] S. Ci, M. Guizani, and G.B. Brahim. A dynamic resource allocation scheme for delay-constrained multimedia services in CDMA 1xEV-DV forward link. *IEEE Journal on Selected Areas in Communications*, 24(1):46–53, 2006.

[15] R. Cruz. Quality of service guarantees in virtual circuit switched networks. *IEEE Journal on Selected Areas in Communications*, 13(6):1048–1056, 1995.

[16] L. David, F. Cottet, and N. Nissanke. Jitter control in on-line scheduling of dependent real-time tasks. In *Real-Time Systems Symposium, 2001. (RTSS 2001). Proceedings. 22nd IEEE*, pages 49–58, 2001.

[17] J. Davin and A. Heybey. A simulation study of fair queueing and policy enforcement. *Computer Communications Review*, 20:23–29, October 1990.

[18] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. *Internetworking Research and Experiments*, 1:3–26, 1990.

[19] D. Eckhardt and P. Steenkiste. Effort-limited fair (ELF) scheduling for wireless networks. In *IEEE INFOCOM*, pages 1097–1106, 2000.

[20] M. El-Kadi, S. Olariu, and H. Abdel-Wahab. A rate-based borrowing scheme for QoS provisioning in multimedia wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 13:156–166, January 2002.

[21] E.S. Elmallah and M. Mandal. Predictive call admission control for multimedia streaming in CDMA cellular environments. *Canadian Journal of Electrical and Computer Engineering*, 30(2), Spring, 2005.

[22] B. Epstein and M. Schwartz. Reservation strategies for multimedia traffic in a wireless environment. In *45th IEEE Vehicular Technology Conference (VTC'95)*, July 1995.

[23] T. Erlebach and F. Spieksma. Interval selection: Applications, algorithms, and lower bounds. *Journal of Algorithms*, 46:27–53, 2003.

[24] K. Etemad. *CDMA2000 Evolution: System Concepts and Design Principles*. John Wiley, 2004.

[25] H. Fattah and C. Leung. An overview of scheduling algorithms in wireless multimedia networks. *IEEE Wireless Communication*, pages 76–83, October 2002.

[26] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the theory of NP-Completeness*. W. H. Freeman and Company, 2003.

[27] S. J. Golestani. A self-clocked fair queueing scheme for high speed applications. In *IEEE INFOCOM*, pages 636–646, April 1994.

[28] P. Goyal, H. M. Vin, and H. Cheng. Start-time fair queueing: A scheduling algorithm for integrated services packet switching networks. *IEEE/ACM Transactions on Networking*, 5:690–704, October 1997.

[29] U. I. Gupta, D. T. Lee, and J. Y. T. Leung. Efficient algorithms for interval graphs and circular-arc graphs. *Networks*, pages 459–467, 1982.

[30] H. Holma and T. Toskala. *WCDMA for UMTS: Radio Access for Third Generation Mobile Communication*. John Wiley, 2001.

[31] D. Hong and S. Rappaport. Traffic model and performance analysis for cellular mobile radio telephone systems with prioritized and nonprioritized handoff procedures. *IEEE Transactions on Vehicular Technology*, 35:77–92, 1986.

[32] J. Hou and Y. Fang. Mobility-based call admission control schemes for wireless mobile networks. *Wireless Communications and Mobile Computing*, 1:269–282, 2001.

[33] R. Howard. *Dynamic Probabilistic Systems*. New York, Wiley, 1971.

[34] J. Hsiao and C. Tano. An efficient algorithm for finding a maximum weight 2-independent set on interval graphs. *Inforation Processing Letters*, 43(5):229–235, 1992.

[35] V. Huang and W. Zhuang. QoS-oriented packet scheduling for wireless multimedia CDMA communications. *IEEE Transaction on Mobile Computing*, 3:73–85, January-March 2004.

[36] International Telecommunication Union (ITU). International mobile telecommunications 2000 (IMT-2000). http://www.itu.int/home/imt.html, 2000.

[37] A. Jamalipour. *The Wireless Mobility Internet: Architectures, Protocols and Services*. John Wiley and Sons Ltd, England, 2003.

[38] L. Jorguseski, E. Fledderus, J. Farserotu, and R. Prasad. Radio resource allocation in third-generation mobile communication systems. *IEEE Communications Magazine*, pages 117–123, February 2001.

[39] N. Joshi, S. Kadbas, S. Petel, and G. Sundaram. Down link scheduling in CDMA data network. In *MobiCom 2000*, August 2000.

[40] P. Kakl. Code excited linear prediction coding of speech at 4.8 kb/s. Technical Report 87-36, INRS-Telecomaaeications, July 1987.

[41] D. Karger, C. Stein, and J. Wein. Scheduling algorithms. In Mikhail J Atallah, editor, *Algorithms and Theory of Computation Handbook*, pages 35–1–35–33. CRC Press, 1999.

[42] S. Keshav. *An Enginnering Approach to Computer Networking: ATM Networks, the Internet, and the Telephone Network*. CRC, 1998.

[43] Y. Khamayseh and E.S. Elmallah. A delay bounded approach for streaming services in CDMA cellular networks. In *Proceedings of the 1st ACM International Workshop on QoS and Security for Wireless and Mobile Networks*, 2005.

[44] Y. Khamayseh and E.S. Elmallah. An adaptive non-preemptive scheduling framework for delay bounded traffic in cellular networks. In *Proceedings of the 31st Annual IEEE Conference on Local Computer Networks (LCN), Tampa, Florida,*, pages 81–88, 2006.

[45] Y. Khamayseh and E.S. Elmallah. Admission control framework for delay bounded traffic in cellular networks. In *Submitted to Globcomm07*, 2007.

[46] T. Kwon, Y. Choi, C. Bisdikian, and M. Naghshineh. QoS provisioning in wireless/mobile multimedia networks using an adaptive framework. *Wireless Networks*, 9:51–59, 2003.

[47] V. Lau and S. Maric. Mobility of queued call requests of a new call-queueing technique for cellular systems. *IEEE Transactions on Vehicular Technology*, 47:480–488, May 1998.

[48] R. Leelahakriengkrai and R. Agrawal. Scheduling in multimedia CDMA wireless networks. *IEEE Transaction on Vehicular Technology*, 52:126–239, January 2003.

[49] C. Leong, W. Zhuang, Y. Cheng, and L. Wang. Optimal resource allocation and adaptive call admission control for voice/data integrated cellular networks. *IEEE Transactions on Vehicular Technology*, 55(2):654–669, 2006.

[50] D. Levine, I. F. Akyildiz, and M. Naghshineh. A resource estimation and call admission algorithm for wireless multimedia networks using the shadow cluster concept. *IEEE Transactions on Networking*, pages 1–12, February 1997.

[51] B. Li, S. T. Chanson, and C. Lin. Analysis of a hybrid cutoff priority scheme for multiple classes of traffic in multimedia wireless networks. *Wireless Networks*, 4:279–290, July 1998.

[52] Y. Liu, S. Gruhl, and E. W. Knightly. WCFQ: An opportunistic wireless scheduler with statistical fairness bounds. *IEEE Transactions on Wireless Communications*, 2:1017–1028, September 2003.

[53] S. Lu, V. Bharghavan, and R. Srikant. Fair scheduling in wireless packet networks. *IEEE/ACM Transactions on Networking*, 7(4):473–489, 1999.

[54] S. Lu, T. Nandagopal, and V. Bharghavan. A wireless fair service algorithm for packet cellular networks. In *MobiCom '98: Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, pages 10–20, New York, NY, USA, 1998. ACM Press.

[55] C. Martel. Preemtive scheduling with release times, deadlines, and due times. *Journal of the ACM*, 29(3):812–829, 1982.

[56] H. Montes, G Gomez, R. Cuny, and J Paris. Deployment of IP multimedia streaming services in third-generation mobile networks. *IEEE Wireless Communication*, 9:84–92, October 2002.

[57] Motorola. Technical overview of CDMA 1x EV-DV, white paper. http://www.cdg.org/resources/, 2002.

[58] G. Mullett. *Wireless telecommunications systems and networks*. Thomson Delmar Learning, 2006.

[59] M. Naghshineh and M. Schwartz. Distributed call admission control in mobile/wireless networks. *IEEE Journal on Selected Areas in Communications*, 14:711–717, May 1996.

[60] T. S. Ng, I. Stoica, and H. Zhang. Packet fair queueing algorithms for wireless networks with location-dependent errors. In *IEEE INFOCOM*, pages 1103–1111, 1998.

[61] A. K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The single-node case. In *IEEE INFOCOM*, volume 2, pages 915–924, May 1992.

[62] B. Patil. *IP in wireless networks*. Prentice Hall Professional Technical Reference, 2003.

[63] E. C. Posner and R. Guerin. Traffic policies in cellular radio that minimize blocking of handoff calls. In *Proceedings of the 11th ITC, Kyoto, Japan*, 1985.

[64] R. Ramjee, R. Nagarajan, and D. Towsley. On optimal call admission control in cellular networks. In *IEEE INFOCOM*, volume 1, pages 43–50, 1996.

[65] T. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall, 2002.

[66] J. Schiller. *Mobile Communication*. Addison Wesley, 2003.

[67] H. Schulzrinne, A. Rao, and R. Lanphier. Real time streaming protocol (RTSP). Technical Report RFC-2326, Network Working Group, April 1998.

[68] S. Shakkottai and R. Srikant. Scheduling real-time traffic with deadlines over a wireless channel. *Wireless Networks*, 8:13–26, 2002.

[69] W. Soh and H. Kim. Dynamic bandwidth reservation in cellular networks using road topology based mobility prediction. In *INFOCOMM 04*, volume 4, pages 2766–2777, 2004.

[70] M. Soleimanipour, W. Zhuang, and G. Freeman. Optimal resource management in wireless multimedia wideband CDMA systems. *IEEE Transaction on Mobile Computing*, 1:143–160, April-June 2002.

[71] J. Spinrad. *Efficient Graph Representations*. American Mathematical Society, 2003.

[72] S. Tekinay and B. Jabbari. A measurement-based prioritization scheme for handovers in mobile cellular networks. *IEEE Journal on Selected Areas in Communications*, 10:1343–1350, 1992.

[73] S. Tsao. Extending earliest-due-date scheduling algorithms for wireless networks with location-dependent errors. *IEEE Vehicular Technology Conference*, 1:223–228, 2000.

[74] I. Wakeman. *Congestion Control for Packetised Video in the Internet*. PhD thesis, University of London, 1995.

[75] X. Wang, R. Ramjee, and H. Viswanathan. Adaptive and predictive downlink resource management in next generation CDMA networks. In *INFOCOMM 04*, 2004.

[76] D. Wu and R. Negi. Effective capacity: a wireless link model for support of quality of service. *IEEE Transactions on Wireless Communications*, 2(4):630–643, 2003.

[77] J. Wu, M. Sze, and J. Chung. Uplink and downlink capacity analysis for two-tier CDMA cellular systems. *Proceedings of the INFOCOM '97*, 00:626–633, 1997.

[78] S. Wu, K. Wong, and B. Li. A dynamic admission policy with precision QoS guarantee using stochastic control for mobile wireless networks. *IEEE Transaction on Networking*, 10:257–271, April 2002.

[79] Y. Xiao, P. Chen, and Y. Wang. Optimal admission control for multi-class of wireless adaptive multimedia srvices. *IEICE Transactions on Communication*, E84-B:795–804, April 2001.

[80] L. Xu, X. Shen, and J. Mark. Dynamic bandwidth allocation with fair scheduling for WCDMA systems. *IEEE Wireless Communication*, pages 26–32, April 2002.

[81] F. Yu, V. Wong, and V. Leung. A new QoS provisioning method for adaptive multimedia in cellular wireless networks. In *INFOCOMM 04*, 2004.

[82] G. Zaruba and I. Chlamtac. A prioritized real-time wireless call degradation framework for optimal call mix selection. *Mobile Networks and Applications*, 7:143–151, 2002.

[83] L. Zhang. VirtualClock: A new traffic control algorithm for packet switching networks. *ACM Transactions on Computing Systems*, 9:101–124, May 1991.

[84] D. Zhao, X. Shen, and J. Mark. Radio resource management for cellular CDMA system supporting heterogeneous services. *IEEE Transaction on Mobile Computing*, 2:147–160, April-June 2003.